UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

MULTICLASS BONE SEGMENTATION OF PET/CT SCANS FOR AUTOMATIC SUV EXTRACTION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

CHRISTIAN FAVIO HURTADO EGUEZ

Norman, Oklahoma

2021

MULTICLASS BONE SEGMENTATION OF PET/CT SCANS FOR AUTOMATIC SUV EXTRACTION

A THESIS APPROVED FOR THE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Joseph Havlicek, Chair

Dr. Hong Liu

Dr. Bin Zheng

Acknowledgements

I would like to thank Dr. Joseph Havlicek, my thesis advisor, for his valuable guidance and support through my entire master's program, and for giving me the opportunity to work on multiple research projects. I would also like to thank Dr. Hong Liu and Dr. Bin Zheng, members of the committee, for their time and willingness to review this thesis.

I would like to acknowledge Dr. Chuong Nguyen for providing the annotated data used in this thesis, and MS Brandon Carson for the SUV statistics used for comparison of results.

Finally, I would like to thank Dr. Jennifer Holter-Chakrabarty, MD of the Stephenson Cancer Center, University of Oklahoma Health Sciences Center, and Dr. Kirsten M. Williams, MD of Children's Healthcare of Atlanta and Emory University Winship Cancer Institute for allowing me to work on their HSCT imaging data.

The HSCT imaging project was funded by the National Cancer Institute, NIH, under contract number HHSN261200800001E.

Table of Contents

Abstract

In this thesis I present an automated framework for segmentation of bone structures from dual modality PET/CT scans and further extraction of SUV measurements. The first stage of this framework consists of a variant of the 3D U-Net architecture for segmentation of three bone structures: vertebral body, pelvis, and sternum. The dataset for this model consists of annotated slices from the CT scans retrieved from the study of post-HCST patients and the ^{18}F-FLT radiotracer, which are undersampled volumes due to the low-dose radiation used during the scanning. The mean Dice scores obtained by the proposed model are 0.9162, 0.9163, and 0.8721 for the vertebral body, pelvis, and sternum class respectively. The next step of the proposed framework consists of identifying the individual vertebrae, which is a particularly difficult task due to the low resolution of the CT scans in the axial dimension. To address this issue, I present an iterative algorithm for instance segmentation of vertebral bodies, based on anatomical priors of the spine for detecting the starting point of a vertebra. The spatial information contained in the CT and PET scans is used to translate the resulting masks to the PET image space and extract SUV measurements. I then present a CNN model based on the DenseNet architecture that, for the first time, classifies the spatial distribution of SUV within the marrow cavities of the vertebral bodies as normal engraftment or possible relapse. With an AUC of 0.931 and an accuracy of 92% obtained on real patient data, this method shows good potential as a future automated tool to assist in monitoring the recovery process of HSCT patients.

Chapter 1. Introduction

Medical imaging provides non-invasive means for expert physicians to evaluate and diagnose disease [1]. In this context, image processing is used to facilitate the evaluation process. In recent years, and especially in view of the explosive growth in machine-learning research, convolutional neural networks (CNNs) have been shown effective in a variety of image processing tasks including important medical applications such as segmentation [2]-[6]. Motivated by this fact and by the study performed by Williams et al. on hematopoietic stem-cell transplant (HSCT) patients [7], [8], in this thesis I present a CNN-based framework for automated segmentation of three bone structures: vertebral body, pelvis, and sternum, from dual modality PET/CT scans. Based on these segmentations, I then present an automated SUV extraction method which can be used for monitoring the patient status during the recovery process and for detecting a proper recuperation versus a possible relapse.

1.1. Problem Description

This work is mostly based on the research by Dr. Williams et al. on HSCT patients [7], [8]. In their study, eligible patients presenting leukemia and myelodysplastic syndrome underwent radiation and chemotherapy in order to eradicate the cancerous cells located in the bone marrow. Then, patients received a venous infusion of haemopoietic stem cells to recover normal hemopoiesis on the host. Their study also establishes that the first 28 days post-HSCT are crucial to the patient for achieving a proper recovery and growth of blood cells (viz. engraftment). If the transplantation is rejected, a *graft failure* takes place [9]; even worse, if cancer is recurrent after the

transplant, a *relapse* occurs. During this stage, the general procedure for examination of the patient evolution consists of a bone marrow biopsy, which is an invasive process that can cause pain and discomfort to the patient [10]. As an alternative, dual modality PET/CT imaging has been proposed for monitoring the patients, which consists of a two-stage procedure: CT imaging followed by PET scanning. The SUV measurements obtained from the PET scan are used as an indicator of the metabolic activity within the bone marrow of the vertebral bodies and other organs.

In the study of post-HSCT patients presented in [7], [8], ^{18}F-FLT was used as a radiotracer for the PET scans and scanning was performed on different instances: the day before the transplant (with the bone marrow ablated), between 5 and 9 days post-transplant, and 28 days post-transplant. Since patients are particularly vulnerable after the myeloablative process, the CT scanning was performed using low-dose radiation (120 kVp). As consequence, the obtained CT volumes comprise anisotropic voxels (approx. size 1.17 mm × 1.17 mm × 5 mm) having a low resolution in the axial dimension, which makes the task of identifying each individual vertebra particularly hard, even visually, due to the CT axial slice thickness being on the same order as or even thicker than the thickness of intervertebral discs of the cervical region [11]. On the other hand, the voxels in the obtained PET volumes are isotropic (approx. size of 4 mm × 4 mm × 4 mm), exhibiting a slightly better axial resolution than the CT scans.

The difference in the resolution between the two imaging techniques generates an issue for obtaining the desired SUV measurements, since the data required for calculating the SUV is contained in the PET scans whereas the relatively better resolution of the CT scans within the axial plane makes it desirable to perform bone segmentation on the CT images. Indeed, the axial

slices in the PET modality do not capture the features of internal structures such as bones and organs accurately, especially in early scanning after the HSC transplant when the metabolic activity measured by the radiotracer within the bone marrow cavities is generally low. Additionally, the initial reference point varies when changing from CT to PET scanning, causing a misalignment between the images. As a result, assessment of the obtained data is a sensitive task even for specialists in the area.

In the next section I present an overview of my proposed solution for automatic segmentation of individual vertebral bodies and SUV extraction from the retrieved PET/CT scans, which I will explain thoroughly in this thesis.

1.2. Proposed Solution and Organization

Currently, the data obtained from the post-HSCT study is evaluated by physicians in a time-consuming task where they need to manually identify, locate, and draw multiple regions of interest on each scan using proprietary medical imaging software [12]. To assist physicians in this task, I propose an automated framework consisting of a CNN for segmentation of the bone structures present in the CT scans and an iterative algorithm for identifying individual vertebral bodies. The obtained segmentation masks are then translated to the PET image space to extract the requested SUV measurements and to calculate some statistics of interest for medical analysis. Finally, a CNN-based classifier is used to classify the patterns generated by the spatial distribution of the SUV within the bone marrow of the vertebral bodies, which has been suggested could be used as an indicator of a successful engraftment or relapse after the HSC transplant [12]. The proposed framework consists of the following stages:

a) 3D U-Net for multiclass bone segmentation: motivated by the extended usage of convolutional neural networks over the last years on image processing

tasks, and based on the work presented in [4], I trained a 3D variant of the U-Net architecture [5], [6] for automated segmentation of three bone structures: vertebral body, pelvis, and sternum. The CT scans obtained in the post-HCST study presented in [7], [8] were used for training the network, with ground-truth annotations provided by Nguyen [13]. To overcome the problem of the small size of the dataset, I applied data augmentation during the network training. The implementation details, along with a discussion of the network performance and comparison with other similar works, is presented in Chapter 3.

b) Instance segmentation of vertebral bodies: using the segmentation mask obtained by the 3D U-Net from the previous stage, I developed an iterative algorithm for identifying and labeling each individual vertebra, starting from C2 and moving downwards to L5. The criteria used for identifying the starting point of a vertebra was based on two anatomical priors: the characteristic curvature of the spine when viewed sagitally, and the presence of pedicles that act as a bridge between the vertebral bodies and the transversal processes of the vertebra. The implementation details and the obtained results are discussed in Chapter 4.

c) Conversion of CT masks to the PET image space: to extract the desired SUV measurements from the PET scans, I translated the segmentation masks obtained from the CT scans to the PET image space by using affine matrices [14] containing the spatial information related to each imaging technique. I then used the extracted values to calculate several statistics of interest, including the mean, median, maximum value and standard deviation of the SUV within the bone marrow of vertebral bodies. Section 4.4 covers the image space conversion and a comparison of the obtained SUV statistics with the SUV results presented by Carson [15] for the same HSCT dataset.

d) CNN-based classifier for post-HSCT evaluation: it has been suggested that the patterns generated by the spatial distribution of the SUV within the bone marrow cavities of the vertebral bodies could be used as an indicator of a successful engraftment or relapse after the HSC transplant [12]. Based on this statement, I trained a 3D CNN based on the DenseNet architecture [16] for classifying the spatial distribution of the SUV obtained in Chapter 4 into two categories: "normal" and "irregular" pattern. The implementation details and network performance are presented in Chapter 5.

The rest of this thesis is organized as follows. In Chapter 2, I present background material on medical imaging terminology used throughout the document, along with a review of image segmentation techniques and convolutional neural networks. I also include a literature review of published works related to spine and vertebral body segmentation. In Chapter 3 I provide the architecture and training details of the 3D U-Net model used for multiclass bone segmentation, and a discussion of the network performance on the post-HCST dataset. In Chapter 4, I introduce an iterative algorithm for instance segmentation of vertebral bodies, which is based on anatomical priors of the spine. I also included the methodology used for extracting the SUV measurements from the PET scans using the segmented masks from the CT volumes, and a comparison of the SUV statistics with the similar work presented in [15] for the post-HCST dataset. In Chapter 5 I provide the architecture and training details of the DenseNet model used for classifying the patterns of the spatial distribution of the SUV within the bone marrow of the vertebral bodies, along with a discussion of the network performance. Chapter 6 serves as a conclusion for this thesis, where I list the original contributions of this work along with recommendations for further research.

Chapter 2. Background

This chapter provides a general background on medical imaging and anatomical definitions used during the development of the proposed framework in addition to a review of image segmentation and deep learning in the context of image processing for medical applications.

2.1. Medical Imaging and Anatomical Key Definitions

2.1.1. Computed Tomography

Computed tomography (CT) is an image acquisition method for clinical use which consists in a patient being exposed to, depending on the configuration, sequential X-ray radiation doses along the region of interest (multi-slice CT), or a rotational beam moving around the subject (helical CT). CT imaging provides additional depth information when compared to traditional radiography, thus, resulting in a 3D volume [17]. The equipment for a CT system consists primarily of the patient table, where the patient lies down during the procedure, the X-ray tube emitting the radiation, and detectors, which measure the radiation attenuation after traversing the patient [18]. The actual image can be reconstructed using algebraic approaches (like backprojection), statistical methods or Fourier-based techniques, giving grayscale values expressed in Hounsfield units (HUs) defined by

$$HU(x,y,z) = 1000\,\frac{\mu(x,y,z)-\mu_w}{\mu_w}, \tag{1}$$

where $HU(x,y,z)$ represents the Hounsfield units at location (x,y,z), $\mu(x,y,z)$ is the corresponding average linear attenuation coefficient at the same location, and μ_w is the linear attenuation coefficient for water at the specific conditions used in the procedure [19].

The spatial resolution of the obtained image is determined by factors like focal spot, motion, detector dimensions and sampling. The noise present will depend on radiation dose, exposure time, slice thickness and reconstruction method used. As a consequence, obtaining an isotropic volume (where each voxel dimension is the same in the three spatial axes) with low noise requires a larger exposure time, which could be impractical due to the breathing movement of the patient [20], [21].

2.1.2. Positron Emission Tomography

Positron Emission Tomography (PET) is a nuclear-based imaging technique. The patient receives a small dose (generally injected) of a radioactive substance, called a radiotracer or radionuclide, which is used for detecting the metabolic activity of the cells of body tissues [22]. The basic principle for PET is the spontaneous positron emission by the nuclei of the radiotracer. The positron annihilates with an electron and releases two gamma particles in opposite directions. A detector ring will record when two opposite gamma photons are sensed within a range of coincidence [23], and that information will be used for reconstructing an image.

PET is used in cardiology, neuropsychiatry, and mostly in oncology for identifying tumors, diagnosis of malignancy, response to treatment, and detection of recurrences [24]. A metric for quantitative assessment of PET is the standardized uptake value (SUV) [25], defined by

$$SUV = \frac{Radioactive\ concentration\ in\ tissue\ \left[\frac{Bq}{ml}\right]}{Injected\ dose\ [Bq]\ /\ weight[Kg]}. \quad (2)$$

The resulting units for the SUV are density units, like [g/ml], indicating the ratio between regional and whole-body concentration. These values play a significant role in this thesis, since it has been suggested that SUV may indicate the status of patients with leukemia after treatment [7], [8].

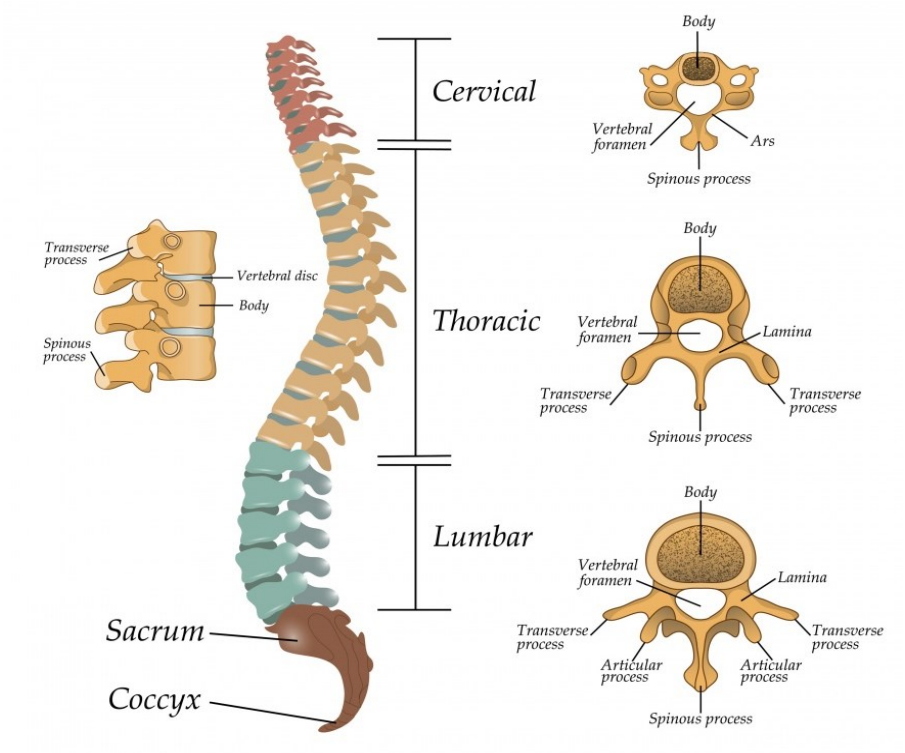


Figure 1. Anatomy of the spine and vertebrae. Extracted from [26].

2.1.3. The Spine

The spine is a column of several stacked bones called vertebrae. It extends from the base of the head to the pelvic zone and serves as support for the human body. The very first group of vertebrae starting from the head are called *cervical vertebrae* (named C1-C7), followed by the *thoracic vertebrae* (T1-T12) and *lumbar vertebrae* (L1-L5) [27]. The sacrum and the coccyx are located at the tail of the spine, both being considered as fused vertebrae. The anatomy of the vertebrae vary for each region, as indicated by Figure 1, the common factor being the presence of a roughly cylindrical vertebral body with some salient structure called the spinous process [28], and the exception being the first two cervical vertebrae C1 (atlas) and C2 (axis).

For the present work, one of the major tasks is detecting and identifying individual vertebrae from CT scans.

2.1.4. Haemopoietic Stem Cell Transplantation

Hematopoiesis or hemopoiesis is the biological process, originated by the hematopoietic stem cells (HSC), in which blood cells are generated [29]. This process occurs primarily in the bone marrow, which constitutes the soft tissue of the bones, corresponding mainly to the interior of the vertebral body for the spine [30].

Some diseases like leukemia may affect the normal production of blood cells, putting the life of the patient at risk. HSC transplantation (HSCT) is a medical procedure in which HSC cells are transplanted to the patient after ablating the immune system with chemotherapy and/or radiation [31]. An *engraftment* takes place when the normal activity of the patient is restored, and *graft failure* is when transplantation is rejected [9]. If cancer is recurrent after the transplant, a *relapse* occurs. PET imaging is used for monitoring the patient's metabolism after the treatment. Although ^{18}F-FDG is the common radiotracer used for this purpose [32], ^{18}F-FLT has been proposed as an alternative because it seems to better capture the metabolism in the bone marrow [7], [8], [33]. Williams et al. performed a study on 23 HSCT patients and scanned the subjects using CT and PET imaging with ^{18}F-FLT [7], [8]. Imaging was executed on multiple instances: one day before, between five and nine days after, and 28 days after the transplant. The study of the obtained scans constitutes the major basis for this thesis.

2.1.5. Anatomical, World and Image Coordinate Systems

The anatomical coordinate system is used for describing the patient's position. It consists of three planes [34] perpendicular to each other:

- Sagittal plane: Divides the body into left and right sections. When the body is split into two equal halves, it is called median sagittal plane.

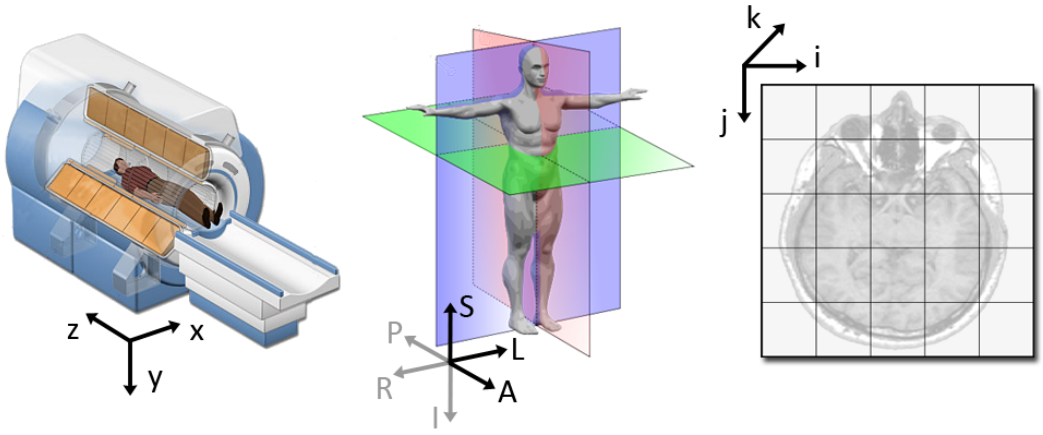


Figure 2. From left to right, illustration of world, anatomical and image coordinate systems, extracted from [35].

- Coronal plane: Splits the body into front (anterior) and back (posterior).
- Axial plane: Also known as traversal plane, divides the body into superior (towards the head) and inferior (towards feet) sections.

The prior definitions are used for specifying the imaging reference planes: Superior/Inferior (S-I), Anterior/Posterior (A-P), Left/Right (L-R). In radiography, the reference point is the patient's soles, resulting in an LPS+ system: positive values defined from right towards left, from anterior towards posterior, and from inferior towards superior on the sagittal, coronal, and axial planes respectively [14], [35]. Other medical applications may use different reference systems.

The world coordinate system is a cartesian system relative to, as the name suggests, a real-world reference point, and is expressed in measurable units (like mm). The reference point may vary from fabricant to fabricant, application, etc.

The image coordinate system is an index-based system, used for representing the actual image data which is stored as an array. In 3D imaging

systems, each voxel (i, j, k) represents the intensity values, and the voxel dimensions represent displacements on the real-world coordinates [35]. Figure 2 illustrates the world, anatomical, and image coordinate system.

2.1.6. Affine Transformation

The correspondence between the image and real-world coordinate systems is given by an affine transformation. On an $\mathbb{R}^n$ space, an affine transformation is a map $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ of the form

$$F(\boldsymbol{p}) = M\boldsymbol{p} + \boldsymbol{q} \tag{3}$$

for all $\boldsymbol{p} \in \mathbb{R}^n$, where $\boldsymbol{q} \in \mathbb{R}^n$ is the translational part of F and M is a linear transformation of $\mathbb{R}^n$, also called the linear part of F [36].

The correspondence between an image voxel (i, j, k) and a real-world coordinate is given by

$$\boldsymbol{p} = \boldsymbol{M}(i, j, k)^{\boldsymbol{T}} + \boldsymbol{p_0}, \tag{4}$$

where $\boldsymbol{p}$ represents a real-world point (x, y, z), $\boldsymbol{p_0}$ is the origin (x_0, y_0, z_0) and M is a 3×3 transformation matrix, originated by the product between the image scaling (also known as spacing or zooming) S and rotations (R_i, R_j, R_k) applied to the image around two axis according to

$$M = SR_iR_jR_k$$
$$= \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{bmatrix} \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where θ, ϕ and γ are the rotational angles with respect to each image axis [36]. The translational part $\boldsymbol{p_0}$ from Eq. (4) can be included in the linear part by using an augmented 4×4 matrix A, generating what is called homogeneous coordinates [37], as indicated by the following expression:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = A \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & x_0 \\ M_{21} & M_{22} & M_{23} & y_0 \\ M_{31} & M_{32} & M_{33} & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix}. \tag{6}$$

This reduces the conversion from real-world coordinates to the image space to

$$\begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix} = A^{-1} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{7}$$

In case of switching between two different image spaces, the voxel coordinates from image space B containing the same spatial position as the initial one can be found by

$$\begin{bmatrix} i' \\ j' \\ k' \\ 1 \end{bmatrix} = B^{-1} A \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix}. \tag{8}$$

2.1.7. DICOM Data Format

DICOM stands for "Digital Imaging and Communications in Medicine." DICOM is an international standard used in medical imaging, originated in the 1980s by the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) for generating a unified format shared between fabricants of different imaging devices (CT, PET, fluoroscopy, angiography, etc.) [38]. The current standard is based on the third revision from 1993, and updates are being released periodically. The standard defines the structure for storing imaging and patient data and additionally a communication protocol for exchanging information.

For 3D images, the DICOM format uses a slice-by-slice basis for storing data, which means that for a single patient scan, multiple files are generated, each representing an axial slice. Each file contains the image data, stored as a

row-column array, and a metadata dictionary, which includes the patient's age, weight, height, and procedure. The dictionary also stores the spatial information of the image frame, such as rotation, pixel spacing, slice thickness and position in millimeters, as well as binary-related information such as bit-size, data type and number of channels [39].

Additional information is recorded depending on the procedure being used. For CT imaging, this includes the current, voltage and exposure time; for PET imaging, the additional information includes the radiotracer, dose, and decay time.

2.1.8. NIfTI Image Format

The Neuroimaging Informatics Technology Initiative (NIfTI) was founded in the 2000s by the National Institute of Mental Health (NIMH) and the National Institute of Neurological Disorders and Stroke (NINDS). The NIfTI format extension (.nii) was designed as a relatively simple storage format for neuroimaging. The current revision NIfTI-2, approved in 2011, allows 64-bit storage [40].

The image data in a NIfTI file is stored sequentially as a list, representing a whole volume, or a time series volume for some procedures. The file header stores the total dimension of the image, the data type, bits per pixel, and voxel units. The spatial information is stored in the form of an affine matrix.

Although the NIfTI format was initially designed for working with magnetic resonance imaging, radiological data such as that present in a DICOM file for CT scan can also be stored with the proper considerations [41].

2.2. Image Segmentation

One of the common tasks in image processing is image segmentation, which is the process of partitioning an image into multiple segments or regions, each

one with homogeneous characteristic features such as texture, morphology, brightness, etc. Each segment S_i is associated with a label, and each pixel (or voxel in 3D images) belonging to S_i is assigned with the same label value [42]. If the segmentation goal is to identify non-countable or general regions or classes (such as sky, car, people), it is referred as *semantic segmentation*. Instead, when it is desired to identify individual objects or instances (like each one of the cars captured by a CCTV cam), the term *instance segmentation* is used. A combination of both semantic and instance segmentation is known as *panoptic segmentation* [43]. Unless otherwise specified, I will use the term segmentation when referring to semantic segmentation.

2.2.1. Segmentation Techniques

Multiple techniques have been studied and developed for image segmentation. The most basic algorithm is thresholding, which consists of converting a grayscale image to a binary image by clipping the values below or above a reference level (or threshold). This method is useful when there is a high contrast between background and foreground, or when the objects to be segmented each present similar intensity values that are distinct from one another [44]. Adaptive thresholding techniques are based on local thresholding and they usually make use of the statistics of a subregion such as mean, median, or peak values.

Edge-based techniques are used for detecting the edges or boundaries of an object. An edge is considered to be a discontinuity in the intensity values between two regions, which can be detected using discrete spatial filters based on the first (gradient) and second order (Laplacian) derivatives. Common filters include the Robert, Sobel, and Laplacian of Gaussian kernels and the Canny operator [45].

Region based segmentation, as the name suggests, makes use of subregions within the image. Some examples include:

- Region growing: a technique that groups pixels into larger regions by using the criteria defined on the initial seed points.

- K-means clustering: splits the image into multiple clusters and runs iteratively until the variance within clusters is minimized.

- Graph cut segmentation: represents the image as a graph, where the nodes represent pixels which are connected by edges [45].

Model-based segmentation represents the shape and structure of an object by some algebraic or geometrical model [42]. Template matching compares the features of a target region with a pattern. This procedure was used in [46] for tracking down markers on tumors and in [47] for segmentation of cells. Parametric deformable models represent the contours as a parametrized curve affected by internal and external forces that define the object boundary [42], [48], [49].

In recent years, machine learning and, more specifically, deep-learning methods have regained strength due to the increase in computational power and GPU memory size. A more detailed discussion is presented in Section 2.3.

2.2.2. Metrics for Image Segmentation

Segmentation can be considered a classification task, in which each voxel or pixel is assigned a label or value representing a class (background or foreground in the case of binary segmentation) [45]. Thus, most of the following concepts also apply to image classification. A fuzzy segmentation algorithm, like deep-learning-based segmentation, assigns weights or probabilities in the range [0,1], which are converted to discrete values by using some criteria [50],

Table 1. Confusion Matrix.

	Actual Positive	**Actual Negative**
Assigned Positive	True Positive (TP)	False Positive (FP)
Assigned Negative	False Negative (FN)	True Negative (TN)

[51], like thresholding. The segmentation results are then tabulated in the form of a confusion matrix, as shown in Table 1. The values in the confusion matrix are determined by comparing the segmentation results with the ground truth values. The ground truth values, or actual values, are a set of annotations containing the ideal or expected result [52]. True positive and true negative values indicate agreement between the label assigned to a pixel or voxel and the ground truth. A mismatch on the assigned label generates the false positive and false negative values. Using these values, the following metrics can be calculated:

- *Sensitivity:* also called *recall* and *true positive rate*, represents the ability to predict positive values. It is given by

$$TPR = \frac{TP}{TP+FN}. \tag{9}$$

- *Specificity:* true negative rate, represents the ability to predict negative values. It is given by

$$TNR = \frac{TN}{TN+FP}. \tag{10}$$

- *Fallout:* the false positive rate, given by

$$FPR = \frac{FP}{FP+TN}. \tag{11}$$

- *Miss rate:* the false negative rate. As the name suggests, it represents the incapacity to detect positive cases. It is given by

$$FNR = \frac{FN}{FN+TP}. \tag{12}$$

There is a correspondence between the above expressions which allows one to represent the *FPR* and *FNR* metrics in terms of the other two according to

$$FPR = 1 - TNR, \tag{13}$$

$$FNR = 1 - TPR. \tag{14}$$

Therefore, only two of the previous metrics are necessary for characterizing a model and generating a Receiver Operating Characteristic (ROC), which is a graphical representation for comparing classifiers [53].

Another metric of interest is the accuracy, which indicates the degree of agreement between the model and ground truth values [54]. It is calculated by

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \tag{15}$$

The numerator of (15), even if it takes into account the "correct" number of predicted values, is considered a biased metric because the true positive and true negative values are dependent on the criterion used for discretizing the results [53].

Precision (symbolized *PPV*) is the ratio between correct and predicted positive values:

$$PPV = \frac{TP}{TP+FP}. \tag{16}$$

Although precision is rarely used explicitly for evaluating a segmentation model, it serves for defining the Dice coefficient, also known as the Dice score or overlap index [51], which is obtained as the harmonic mean of precision and recall according to

$$DICE = \frac{2TP}{2TP+FP+FN}. \tag{17}$$

The Jaccard index is given by the intersection over the union of the positive values [51]. It is defined by the formula

$$IoU = \frac{TP}{TP+FP+FN}. \tag{18}$$

The Jaccard index can also be expressed in terms of the Dice score according to

$$IoU = \frac{DICE}{2-DICE}. \tag{19}$$

There are many other metrics for evaluating image segmentation. These include distance-based and volume-based metrics [50], [51]; however these will not be discussed in this section since they are not suitable for comparing the results given in this thesis because they are not commonly used for evaluating medical image segmentations in the literature.

2.3. Deep Learning and Convolutional Neural Networks

Over the last years and with the advances in GPU technology, there has been a significant increase in research on machine learning. Machine learning is a branch of artificial intelligence consisting of algorithms that perform a task or process by using information "learned" in the past and extract relevant information from new data in order to increase performance [55], [56]. Machine learning has multiple applications in industry, medicine, robotics, and finance, among others, and is used in tasks like classification, regression, denoising and speech recognition [55].

Models developed for machine learning include decision trees, genetic algorithms, Bayesian networks, and artificial neural networks, with the last one serving as a basis for deep learning and, by extension, convolutional neural networks [57].

2.3.1. Supervised Learning

There are some distinctions in machine learning with respect to the available data and how the learning process occurs. In *supervised learning*, a set of inputs are provided along with annotated outputs or targets, indicating the desired response [55]. The goal is to find a mapping between the two to generate a prediction for new entries. On the other hand, in *unsupervised learning* the targets are not explicitly provided, and the aim of the learning process is to deduce characteristic features from the inputs [56], [58]. A hybrid approach called *semi-supervised learning* is based on the previous two [58]. In *reinforcement learning*, trial and error is used for solving a task in an optimal way [57].

Supervised learning is the most relevant for classification and segmentation tasks in image processing [55]. However, some major problems that may appear with this approach are overfitting and underfitting. Overfitting is caused when the predictive model captures too many features from the available data with the result that it is unable to make a proper prediction when new data with some variations is presented [59]. To address this issue, the data is split into three subsets: training, validation, and test data. Training data is used for optimizing the model and reducing the error on the predictions. Validation data is used for reducing the complexity of the model, and test data is used for measuring the performance of the model. Underfitting occurs when the model is unable to properly capture the features of the data [60], for example with a short training time or when insufficient data is available.

To deal with these issues, the data is preprocessed before training the model. Preprocessing generally involves thresholding and normalizing values in the range [0,1] [56]. In some cases, the size of annotated data is insufficient

for training a model. To generate diversity, a series of transformations are performed to the original data, including scaling, rotating, shifting, and blurring. This technique is called data augmentation and is particularly useful on supervised learning for medical images [5], since the number of available annotated datasets is often small in comparison to general-purpose image processing tasks [61].

2.3.2. Artificial Neural Networks

The concept of Artificial Neural Networks (ANN) dates from 1943, with the introduction of neural units by Warren McCulloch and Walter Pitts [62]. An ANN is a mathematical model resembling a simplified version of the neural activity from the brain: when a stimulus is received, the neurons become active and start building an electrical charge. If a threshold is reached, a pulse is generated and propagated to other neurons [63], [64]. Similarly, an ANN consists of neural units or nodes, organized by layers. When a layer receives an input, it is evaluated, and if a threshold is reached, the data is passed on to other layers of the ANN [56], [60]. Mathematically, the output $\boldsymbol{y}$ for a vector input $\boldsymbol{x}$ received by a layer is given by

$$\boldsymbol{y} = \phi(\boldsymbol{w}^T \boldsymbol{x} + b), \tag{20}$$

where $\boldsymbol{w}$ is a vector of weights containing the learned parameters of the ANN, ϕ is the activation function, generally non-linear, which serves as a threshold, and b is the learned bias for shifting the activation function [56], [57]. As shown in Figure 3, some examples of activation functions include the sigmoid, hyperbolic tangent, and rectified linear unit (ReLU) [56], [60], [66].

The training stage for an ANN consists of successive iterations called *epochs*, where the training data is "fed" to the network for making predictions which are compared to an error function (also called the cost or loss function).

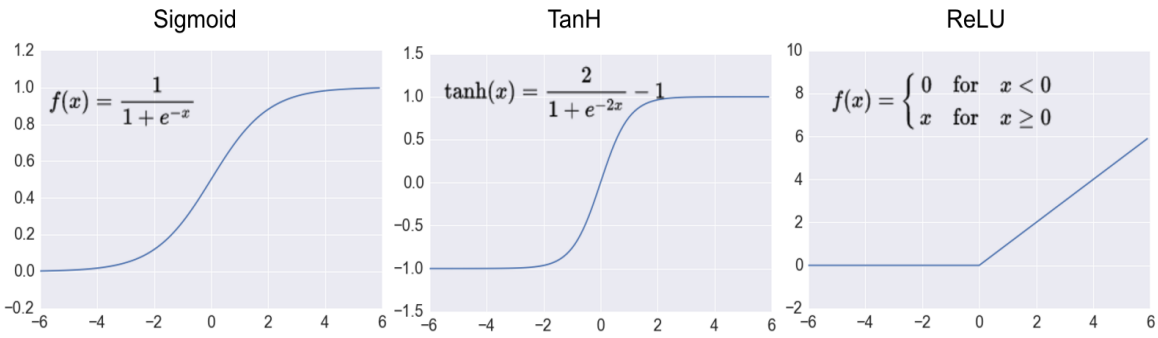


Figure 3. Examples of activation functions for ANNs, extracted from [66].

The weights are then updated to minimize the error using an optimizing algorithm (optimizer) [55], [64].

2.3.3. Deep Neural Networks

Deep neural networks, also known as multi-layer perceptrons [64], consist of multiple layers of neural units interconnected in varied ways. The first and last layers are known as the input and output layers, respectively, and the intermediate ones are known as *hidden layers* because their outputs are not exposed outside the network. The number of hidden layers determines the depth of the network. ResNet, a neural network architecture popularly used for image classification, presents several depth variants ranging from 20 to 1202 hidden layers [65].

With such complex structures, training a network becomes a time and memory consuming task. Recalling from the previous section, the goal of training is to find the optimal parameters θ (weights and biases) that minimize an error function $E(\theta)$. Examples of error functions that are widely used for this purpose include the mean absolute error (MAE), mean square error (MSE), and cross entropy loss (negative log likelihood) [56], [57], [67]. The selection of the error function will depend on the use case for the network. The minimum of the error function will occur when its gradient $\nabla E(\theta)$ is equal to zero. In most

practical situations, the error function is characterized by the presence of numerous local minima, making it infeasible to find an analytical solution for the global minimum [56], [57]. Instead, the gradient descent approach is used, consisting in updating iteratively the weight values by moving small steps towards the greatest rate of descent until convergence occurs. In practice, a stochastic approach called stochastic gradient descent (SGD) is often used which updates the parameters iteratively according to [56]

$$\boldsymbol{\theta}^{(\tau)} = \boldsymbol{\theta}^{(\tau-1)} - \eta \nabla E_n\left(\boldsymbol{\theta}^{(\tau-1)}, \boldsymbol{x}, \boldsymbol{y}\right), \tag{21}$$

where E_n is an error function based on maximum likelihood for a set of independent samples, $\boldsymbol{\theta}^{(\tau)}$ and $\boldsymbol{\theta}^{(\tau-1)}$ are the vector of parameters at the current and previous iterations, $\boldsymbol{x}$ and $\boldsymbol{y}$ are the input and output vectors, and η is the learning rate or step size.

Computation of the gradients is performed by using the *backpropagation* algorithm, which consists of calculating the outputs and errors of all the nodes in all the layers and then propagating the errors from the last layer to the previous ones. Backpropagation is explained in great detail in [56] and [57].

Choosing the proper step size is fundamental for SGD: if the value is too small, then convergence can be slow; if it is too large, the SGD may fail to converge or might even diverge. Several variations for the SGD have been proposed that include additional terms with adaptive parameters adjusted on each iteration. These include SGD with momentum, RMSProp, Nesterov's momentum, AdaGrad, and the Adaptive Moment Estimation Algorithm (Adam) [56], [68]-[70]. To date, Adam is the most widely used algorithm for training neural networks for image processing [56].

An issue that arises when training a deep neural network is that the distribution of each layer's inputs changes as the parameters from previous

layers are updated. This phenomenon is known as *internal covariate shift* and causes primarily a slowdown in the training process [71]. One solution for addressing this issue is the *batch normalization*, consisting in normalizing the outputs of the layers to have zero mean and unit variance, and then applying a linear transformation. Batch normalization has been shown to speed up training and achieve better performances [72].

2.3.4. Convolutional Neural Network

Convolutional neural networks (CNNs or ConvNets) represent the current state-of-the-art in deep neural networks for image processing. CNNs attempt to emulate the functioning of the biological visual cortex [57], [63], similar to the way that ANNs seek to emulate biological neural activity. Since images are multidimensional grids by nature, vectorizing them is not appropriate because of their significant size and the fact that neighborhood information is lost. Instead, it is preferred to process subregions to extract *feature maps* [55], [56], which are significant features of an image such as edges. This can be achieved by convolving the image with a kernel or sliding window instead of using an element-wise multiplication, with the additional benefit of reducing the complexity of the network via *weight sharing*. However, the available libraries for CNN implementation use the cross-correlation operator instead of convolution [73], [74], which serves the same purpose by using symmetric kernels and omitting the "flip" or time reversal that is associated with convolution. The problem of training a CNN then becomes one of finding the optimal weights for the kernels that minimize the error function.

The most common parameters that must be specified in designing a CNN include the following:

- Input channels: the number of stacked feature maps in the input of each layer.

- Output channels: the number of stacked feature maps on the output of each layer.

- Kernel size: the size of the sliding window element. It should be noted that the actual dimensions for a 2D kernel of size 3, with input channels M and output channels N, is $M \times 3 \times 3 \times N$ [75].

- Strides: adjust the step size for the kernel. As a consequence, the outputs are downsampled with respect to the input, thus reducing dimensionality.

- Padding: extends the dimension of the input around the border to handle edge effects.

- Pooling strategy: also reduces the dimensionality of the outputs, by performing an operation (sum, average, maximum) on subregions of the feature map.

A common configuration found in CNNs is the *encoder-decoder* architecture [76], [77]. The *encoder* path consists of a succession of convolutional (properly, cross-correlation) and downsampling layers. Analogously, the *decoder* path consists of a succession of deconvolution and upsampling layers, reconstructing the original size.

2.3.5. CNNs for Medical Imaging Segmentation

The initial approach used to apply CNNs for image segmentation consisted of stacking several convolutional layers [75], [78]. The reasoning behind this was that increasing the number of layers also increases the number of features obtained from the image; however, this makes the training process slow due to the high number of learnable parameters. An example of this kind of architecture was proposed by Cireşan et al. [78], who obtained the first place

prize in the “2012 ISBI challenge for segmentation of neuronal structures in electron microscopic stacks” [79].

Later, Ronneberger et al. proposed a new architecture called U-Net. The U-Net architecture is shown in Figure 4, where the number of input channels is doubled on each downsampling stage of the encoder path and halved in the corresponding upsampling stages [5]. The *copy and crop* connections, also known as *skip connections*, are used for concatenating the corresponding feature maps from the encoder and decoder paths. A 3D variant of the U-Net, called 3D U-Net, was proposed by Çiçek et al., where all layers are replaced by their 3D counterparts [6], as shown in Figure 5. Other major refinements to the original design include the addition of batch normalization after each convolution and using a weighted softmax loss function which allows the network to be trained with sparse annotations [6]. Another 3D variant of the U-Net is the V-net, due to Milletari et al., which uses a parametrized version of the ReLU (viz. PReLU) as the activation function and replaces the pooling layers with stride convolutions [80].

Multiple variants of the U-Net architecture and fully-CNNs have been proposed for medical applications, demonstrating their usefulness for segmentation of the heart [81]-[83] and location of tumors and lesions in the liver [84], [85], brain [86]-[88], and lung [89]-[91].

2.3.6. CNNs for Classification

One of the most popular architectures for image classification is the ResNet, which is characterized by introducing residual units between the convolutional layers [65]. Residual units consist of adding the output of a block with its original input through a shortcut connection. This connection, also known as an identity connection, is used to preserve significant features that may be

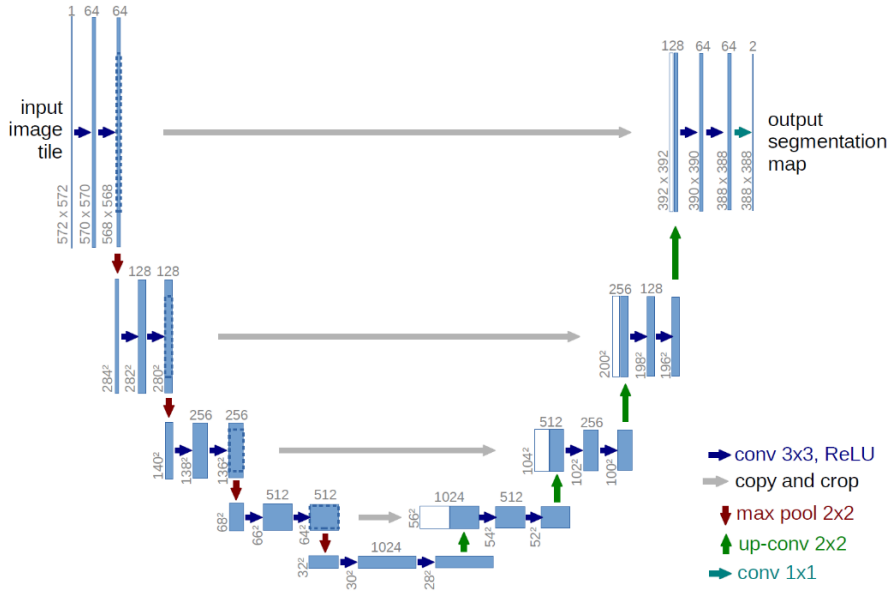


Figure 4. U-Net architecture proposed in [5].

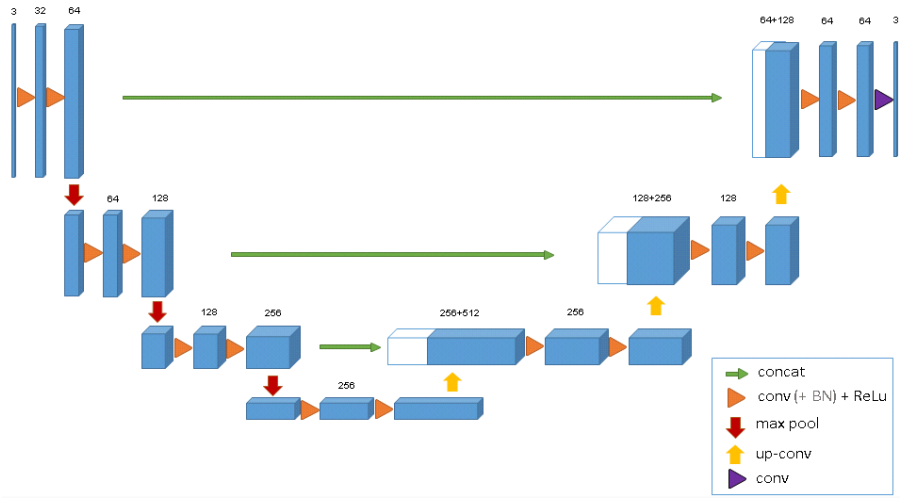


Figure 5. 3D U-Net architecture proposed in [6].

degraded by successive stages. Mathematically, the residual connection can be expressed as [65]

$$\boldsymbol{y}_l = F_l(\boldsymbol{y}_{l-1}, \boldsymbol{w}) + \boldsymbol{y}_{l-1}, \tag{22}$$

where $\boldsymbol{w}$ represents the weights and $\boldsymbol{y}_l$ and $\boldsymbol{y}_{l-1}$ are the current and previous block outputs, respectively. F_l represents a composite function performed by the block, such as convolution, batch normalization, and activation.

Other relevant architectures for classification include VGGNet and DenseNet. VGGNet consists of an encoding path followed by a fully connected layer of 4096 channels [92]. DenseNet, which stands for Dense Convolutional Neural Network, proposes connecting all the layers with the subsequent ones in a block [16]. Unlike ResNet, concatenation is used for the output instead of addition, which may be expressed as

$$\boldsymbol{y}_l = F_l([\boldsymbol{y}_0, \boldsymbol{y}_1, \dots, \boldsymbol{y}_{l-1}]), \tag{23}$$

where the brackets represent the concatenation operation and the $\boldsymbol{y}_i$ represent the outputs of preceding layers [16].

Multiple variations of the mentioned architectures have been evaluated for medical image classification. In [93], the authors trained four different networks, including VGGNet and ResNet, for detecting calcifications in mammography. All the networks presented an overall accuracy over 90%, with VGGNet on top by a slight difference. Other applications for CNN classifiers include analysis of lung [94]-[96], heart [97] and cancer [98], [99] images. A detailed performance comparison of different CNNs can be found in [100].

2.3.7. Frameworks for Deep Learning

Up through the early 2000's, deep learning development relied on relatively simple libraries which did not provide sufficient flexibility [101] and suffered from memory constraints due to hardware limitations.

In 2012, the Caffe [102] framework was released, containing specialized routines for deep learning specifically tailored for image processing. This framework gained wide popularity amongst the research community and it is

still found today embedded in other frameworks [103]. With the advances in hardware technology, more sophisticated libraries were released, including the following which are all widely used:

- MATLAB: the Deep Learning Toolbox is a framework for deep learning with optimization for GPU and parallel processing [104], which when combined with all the tools provided by MATLAB makes it valuable for scientific research.
- TensorFlow: initially developed by Google, TensorFlow provides a relatively simple but powerful API for designing, training, and testing neural networks [74]. It gained popularity in part due to Google Colab, which allows code to be run on remote servers, thereby providing a viable and widely used tool for rapid prototyping.
- PyTorch: it is one of the most popular frameworks for deep learning, providing an entire ecosystem of tools built around it [105]. PyTorch presents a robust API using a specialized data structure named tensors [73]. Tensors store data in an array-like manner, as well as gradient information about the interconnected nodes in a layer. Additionally, multiple operations can be performed among tensors without installing additional libraries. However, all these features are expensive in terms of memory and processing power requirements, requiring proper hardware in order to take advantage of all its capabilities.

2.4. Related Work

Segmentation of the spine is a fundamental step in medical image processing for vertebrae localization. However, identifying each individual vertebra is a challenging task due to their complex structure [106].

In 2014, the "2nd MICCAI challenge for spine segmentation" was held to motivate research on vertebrae segmentation. Most of the entries submitted to

the challenge made use of traditional segmentation methods based on geometric or statistical models [107]-[111]. Although very good results were generally obtained, most of the submitted techniques required manual input on some cases or were specific to a spine sub-region. A detailed discussion can be found in [106].

More recently, the "Large Scale Vertebrae Segmentation Challenge" (VerSe) was introduced in two consecutive years (2019, 2020). The dataset for the challenge consists of over 300 CT scans with annotated labels for individual vertebrae [112]-[114]. Machine learning techniques are dominant among the submitted entries, notably including V-Net implementations [115] and multi-view ensemble U-Nets [116]. A complete list of the submitted techniques and their results is provided in [112].

One noteworthy observation about vertebrae segmentation techniques in general is that good performance requires a high spatial resolution, especially in the axial plane. Thick slices do not provide sufficient resolution for effective discrimination of the intervertebral discs, making the segmentation task harder. The scan volumes obtained from Williams et al. [7], [8], which I will hereafter refer to as the HSCT dataset, are characterized by a slice thickness of 5 mm, whereas the slice thickness in the VerSe dataset ranges from 0.6 mm to 2 mm. Another relevant point is that the annotations for the VerSe dataset designate the entire vertebra, whereas detection of just the marrow cavity is important for assessment of post-HSCT patient images. Nguyen et al. developed a framework for automatic segmentation of the marrow cavities [117], [118] which consisted of a graph-cut segmentation for full-body bone extraction, iterative thresholding in the sagittal plane and Kalman filtering for vertebral disc isolation. They reported an average TPR of 0.916 on the HSCT dataset. Using the same data, Carson trained a multi-view ensemble U-Net for

vertebral body segmentation [15], obtaining a mean Dice score of 0.922. The segmented volumes were then registered with the PET data, which presents a slightly higher resolution in the axial plane, to detect the boundaries between vertebrae. The downside of this approach is the dependency on hematopoietic activity, which is not restored until several days after HSCT, and thus is not readily detected by PET imaging on the initial days after transplant. The solution that I propose in this thesis involves training a 3D U-Net for vertebral segmentation and identifying the individual vertebrae from the CT volumes. The next chapters describe the implementation details and results of this approach.

Chapter 3. 3D U-Net for Multiclass Bone Segmentation

Semantic segmentation of bone structures is the initial step in my proposed solution for extracting SUV values from post-transplant PET/CT scans of HCST patients. Motivated by the demonstrated effectiveness of convolutional neural networks on image processing problems, specifically on vertebrae segmentation [112]-[114], and by the availability of multiple libraries and tools for machine learning [73], [74], [119], I trained a 3D variant of the U-Net architecture for segmentation of three classes: spine, pelvis, and sternum. The training data was obtained from the HSCT study performed by Williams et al. [7], [8], with annotations provided by Nguyen [13].

3.1. Implementation Details

3.1.1. Dataset

The dataset I used in this project was obtained from the research by Williams et al. [7], [8]. In their research, the patients went through radiation and chemotherapy for ablation of the bone marrow, eliminating the cancerous cells. After that, hematopoietic stem cells were infused to the patient. Dual-modality PET/CT imaging was performed for monitoring the patients, with ^{18}F-FLT used as a radiotracer for PET. Three PET/CT scans were obtained per patient, with imaging occurring at one day before HSCT, between 5-9 days after HSCT, and at 28 days after HSCT.

A total of 64 scans for 22 different subjects were acquired. The resulting CT scans are anisotropic volumes with a voxel size of 1.17 mm × 1.17 mm × 5 mm and a resolution per axial slice equal to 512 × 512 pixels. These characteristics make the process of identifying the boundaries between vertebrae harder. On the other hand, the PET volumes are isotropic with spacing of 4 mm × 4 mm ×

Table 2. Distribution of Annotated Volumes for the HSCT dataset.

Segmentation Class	Volumes
Vertebral Body Only	14
V. Body + Sternum	5
V. Body + Sternum + Pelvis	16
None	29
Total	**64**

4 mm, resulting in a lower resolution of 144 × 144 pixels per slice, but providing slightly improved resolution along the transverse axis perpendicular to the axial plane.

Ground-truth values for 35 CT scans were provided via voxel-level annotations obtained using semi-automated methods for three bone structures: vertebral body, pelvis, and sternum [13]. These classes only add up to a small fraction compared to the total volume size, and only 16 of the 35 ground-truth volumes contain annotations for all the three classes. On the remaining volumes, the vertebral body class is the most prevalent, followed by the sternum and pelvis, as indicated in Table 2. Something to notice about the "pelvis" class is that it actually contains voxels belonging to the pelvis, sacrum, and coccyx. The nomenclature for this class, although not clinically accurate, was preserved for simplicity.

The patient scans were stored in a dictionary-based format for MATLAB®, containing the CT volumes in Hounsfield units, normalized SUV values from the PET volume, and the metadata from the original DICOM scans. The annotations consist of raw binary data stored on a slice-by-slice basis. Visualization of the volumes and file reading for training the CNN required writing custom-made classes and methods. This, added to the lack of

uniformity of the file naming convention, motivated me to use a more standard format for storing the data. After analyzing the associated metadata, I opted for using the NIfTI format for the following reasons:

- Spatial information does not vary from slice to slice. Pixel spacing, rotation, and orientation remains constant. Only the axial position is incremented in uniform steps equivalent to the slice thickness. This allows for representation of the spatial information in the form of an affine matrix.

- Simplified folder structure. NIfTI data is stored on a volume basis. Thus, only three files are required per patient scan: one for CT, one for PET and one containing the CT ground-truth values. This considerably reduces the total number of files compared to using DICOM format or the provided binary slices.

- Availability of visualization tools (3D Slicer [120], MATLAB®'s Volume Viewer [121]) and reading and writing packages for NIfTI files [122]-[126]. These tools also handle compressed (.gz) NIfTI files, reducing the disk space required for storage.

With the available data, I generated the new NIfTI files. For the CT and PET scans, I used their corresponding metadata from the first slice to obtain the origin coordinates, orientation, and voxel dimensions, generating the affine matrices needed for the NIfTI format. For the CT ground-truth volumes, I used the same affine matrix as for the corresponding CT scan and assigned the following voxel values based on the annotated slice information: 0 for background, 1 for vertebral body, 2 for pelvis, and 3 for sternum. For each PET/CT scan, I also extracted the most relevant patient and imaging information from the first slice metadata and stored it in the form of a

Table 3. List of metadata attributes extracted from the original dataset files.

Keyword	Data Type	Description
StudyDate	Date	Date the study started
StudyTime	Time	Hour the study started
PatientID	String	Identifier for the Patient
PatientAge	String	Formatted string for patient's age
PatientSize	Numeric	Length of the patient (m)
PatientWeight	Numeric	Weight of the patient (kg)
PatientSex	String	Single char for patient sex (M, F, O)
Modality	String	Coded string for imaging modality
Rows	Numeric	Number of rows per axial slice
Columns	Numeric	Number of columns per axial slice
ImagePositionPatient	Numeric Array	x,y,z real-world coordinates of the first voxel (mm)
ImageOrientationPatient	Numeric Array	Direction cosines of the image orientation
PixelSpacing	Numeric Array	Physical distance between adjacent voxels (mm)
SliceThickness	Numeric	Nominal slice thickness (mm)
KVP	Numeric	Peak voltage output by the X-ray generator (kV)
XRayTubeCurrent	Numeric	X-ray tube current (mA)
Exposure	Numeric	Radiation exposure (mAs)
Radiopharmaceutical	String	Name of the radiopharmaceutical used
RadiopharmaceuticalRoute	String	Method used to administrate the radiopharmaceutical
RadiopharmaceuticalVolume	Numeric	Volume administered (cm^3)
RadionuclideTotalDose	Numeric	Radiopharmaceutical dose administered to the patient (Bq)

spreadsheet. It is worth mentioning that, for PET imaging, the DICOM standard defines many other attributes related to the dose and acquisition time that affect the pixel representation of each slice [39]; however, these were not considered because, as I mentioned earlier, the provided PET volumes were

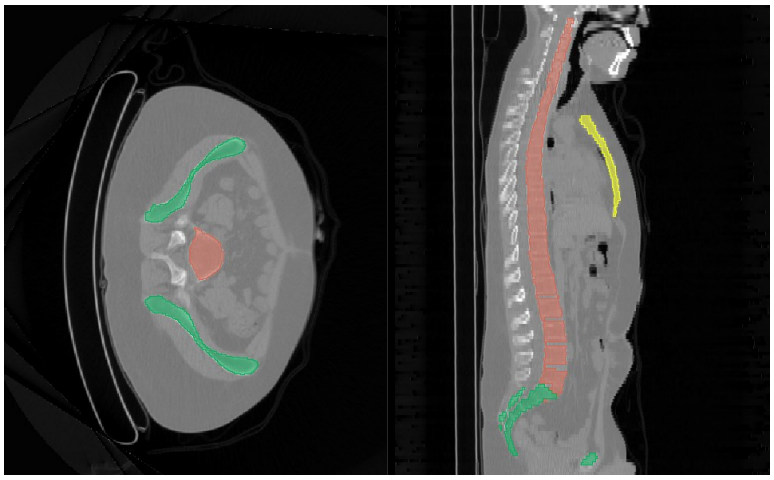


Figure 6. Example of an axial (left) and sagittal (right) CT slices overlapped with their corresponding ground-truth annotations: Vertebral body (red), sternum (yellow) and pelvis (green).

already expressed in SUV units [7]. Table 3 summarizes the selected attributes from the original metadata. An example of the resulting annotations can be seen in Figure 6. In Section 3.1.3 below I will describe how the new generated files were used for training the CNN.

3.1.2. Model Architecture

For the bone segmentation method proposed in this thesis, I selected a CNN architecture based on the U-Net variant introduced by Kerfoot et al. for ventricle segmentation [4]. Their design has residual units in both the encoder and decoder paths and incorporates instance normalization [127] after each convolution. Instance normalization is similar to batch normalization [71] with calculations performed on a per-image level [127]. However, I opted for batch normalization over instance normalization since the chosen batch size in my application is small due to memory constraints. This is unlike the original implementation from [4] where a batch size of 1,200 was used.

For the implementation, I used the PyTorch-based framework MONAI [119], which introduces a stride convolution on the residual unit for matching

input sizes when required. The input for the network consists of four input channels, one for each label class, of size 96 × 96 × 96, which is sufficiently large for preserving spatial information without being too demanding for the hardware. As indicated in the original U-Net paper [5], the number of channels is increased after each block on the encoder path while the spatial size is reduced. In the decoder stage, the inverse process takes place: the number of channels is reduced, and the spatial size is expanded sequentially. The last layer consists of a convolutional layer for retrieving the probability map for each voxel. The kernel dimensions are 3 × 3 × 3 for convolution and 2 × 2 × 2 for downsampling. The selected activation function is PreLU (Parametric Rectified Linear Unit) [128], which has been noted to improve the performance in segmentation networks compared to the ReLU function. PreLU is defined by [128]

$$f(x) = \begin{cases} x, & x > 0 \\ ax, & x \leq 0 \end{cases} \tag{24}$$

where a is a learnable parameter controlling the slope when the argument is negative. PreLU is equivalent to the ReLU activation function when $a = 0$. The final encoder and decoder blocks are shown in Figure 7. With the model ready, I proceeded to the training stage, which is detailed next.

3.1.3. Training

Using the newly converted NIfTI files, I selected the 16 scans that contained annotations for all the bone structures. From those, 12 were used for training and 4 for validation. For increasing data variability, I applied data augmentation [5] with a random probability to the CT volumes, including random scaling and stretching up to 15%, rotation in the range of -45° to +45°, and random volume flipping. The intensity values were normalized to the range [0,1]. For feeding the data to the network, the images were split into

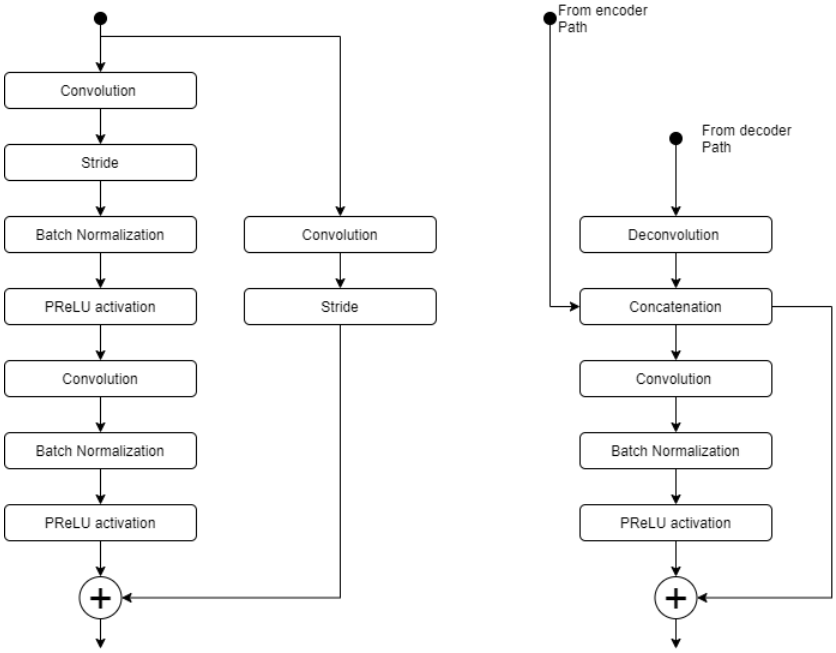


Figure 7. Blocks for the encoder (left) and decoder (right) paths. Adapted from [4] with modifications as described in Section 3.1.2.

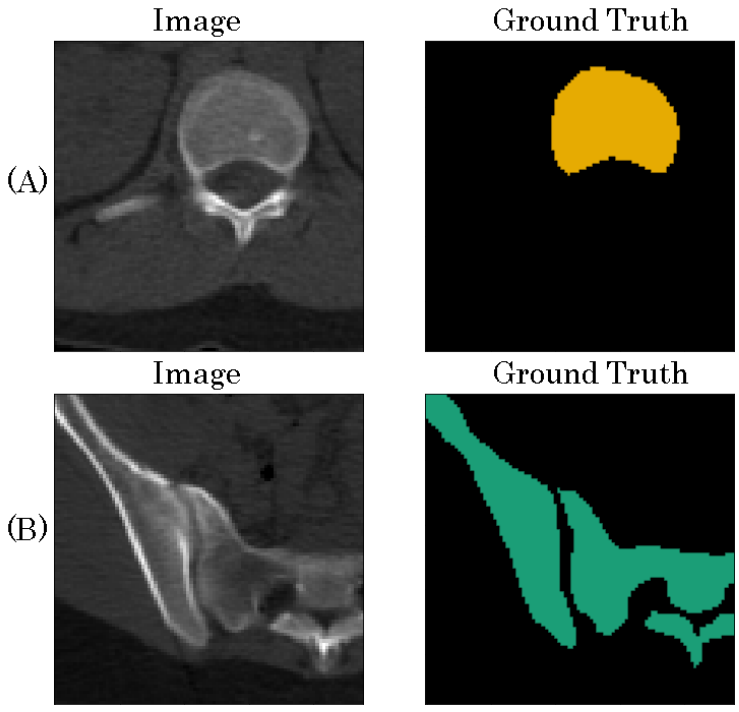


Figure 8. Examples of the image (left) and ground truth mask (right) generated after preprocessing. A: vertebral body. B: pelvis.

multiple subvolumes or *patches* with size 96 × 96 × 96, which were then concatenated to restore the original size. Figure 8 shows some examples of the preprocessing result.

For completing the model training, I used the workflow described in [129]. In the training loop, a mini-batch of data was loaded on each iteration and passed through the network to generate an output, which was then compared to the ground-truth values with the error function (which I describe below). The output was then backpropagated to attempt convergence using the optimizer. In the validation loop, the network performance was evaluated using the validation data. For the optimizer, I chose the ADAM algorithm [70], which requires as parameters the learning rate α and exponential decay rates (β_1, β_2) for the estimates of the first and second moments of the gradient. Using the guidelines from [130], I assigned the values $\beta_1 = 0.9, \beta_2 = 0.999$ and 10^{-4} for the learning rate.

Due to the imbalanced proportion of background and non-background voxels, I opted for using the Dice Loss [80] as an error function. As the name suggests, it is adapted from the Dice score. For a certain volume with N voxels, the Dice Loss is defined by

$$DL = 1 - \frac{2\sum_i^N p_i g_i + \epsilon}{\sum_i^N p_i^2 + \sum_i^N g_i^2 + \epsilon}, \tag{25}$$

where p_i, g_i represent the predicted probability and ground truth for the ith voxel, respectively, and ϵ is a small constant added to avoid division by zero. Since $g_i = 0$ for background voxels, these values do not contribute to the summation and the class imbalance is addressed without the need of assigning weights to the classes, which could potentially introduce some bias to the results.

Figure 9. Dice Loss results over training.

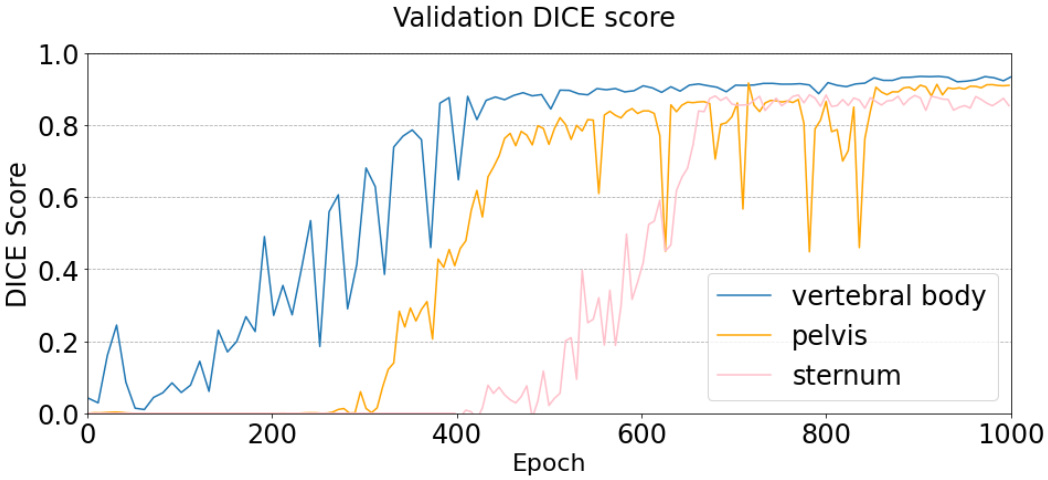


Figure 10. Validation scores during network training.

The training was performed using an NVIDIA® GeForce RTX 2070 graphic card with 8GB of GDDR6 SDRAM. Training ran for 1,000 epochs for 12 scans with a mini-batch size of 2 to prevent exceeding the available memory, resulting in a total number of iterations equal to 6,000 (1,000 epochs × (12 scans /(batch size of 2))). Periodic validation metrics were calculated to monitor the model behavior with new data, storing a snapshot of the model status to disk when performance increased. Figure 9 shows the Dice Loss evolution during training. The trend shows that the model does not diverge and there is an acceptable margin of 0.12 to prevent overfitting. The plot in Figure 10

indicates that the validation scores obtained are above 0.8, with the spine presenting higher values due to its major proportion compared to the other classes. The negative spikes which appear for the pelvis class are attributed to some inconsistency in the annotations, where coccyx was not being included. These volumes were ignored for evaluation to improve reliability of the results.

3.2. Results

3.2.1. Quantitative Results

After training, the model was evaluated with aid of the remaining annotated volumes; however due to a lack of additional annotations for the pelvis class, a fraction of the training and validation volumes was reutilized for this purpose. The evaluation consisted of feeding data to the network to generate a probability map for each one of the voxels, then converting the values to discrete values to generate a confusion matrix for each one of the predefined classes. A straightforward method for doing this is to select a threshold value T to assign the values of 0 or 1 when the probability from the generated map is above or below the reference level according to

$$Y(m,i) = \begin{cases} 1\,, & P(m,i) \geq T \\ 0, & otherwise \end{cases} \tag{26}$$

where $P(m,i)$ represents the generated probability for a voxel i belonging to the class m and $Y(m,i)$ is the thresholded value for said voxel. By choosing two different values for T in the range [0.4,0.5], I generated two confusion matrices as described in section 2.2.2, which were used for plotting the ROC curve shown in Figure 11. It can be seen that for the three classes the area under the curve is greater than 0.9, which gives the initial impression that the model behavior is reasonably close to that of an ideal classifier. However, one must keep in mind that the imbalance of the foreground voxels with respect to the background leads to a low FPR. Consider a patient scan of 512 × 512 × 200

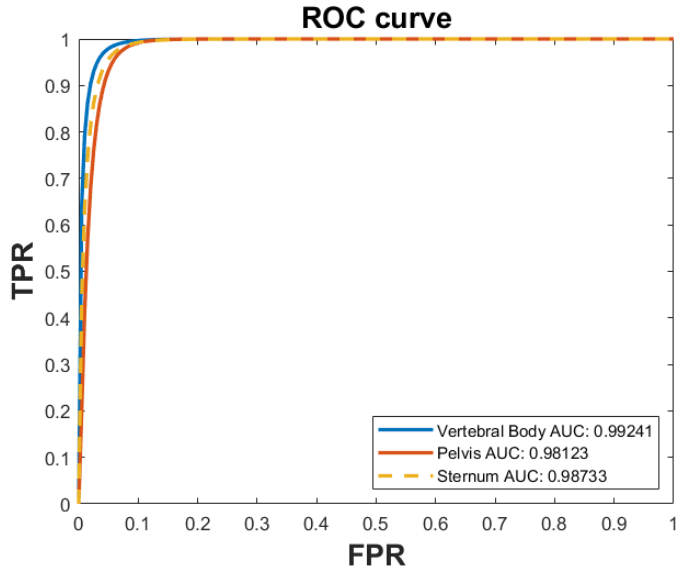


Figure 11. ROC curve and AUC values for the three classes.

voxels. The percentage of voxels occupied by vertebral body, pelvis, and sternum combined is below 1%. An arbitrary choice of the threshold does not impact the magnitude of the true negative values significatively, thus resulting in very low FPR values.

I calculated the Dice Score on the test volumes using a threshold $T = 0.5$, but I encountered two issues with the approach: first, the probability map values for voxels located on the edge of the bone structure are slightly below 0.5, causing missed detections and, moreover, the ideal threshold value varies from volume to volume. Second, overlapping occurs along the boundary between the spine and pelvis where both probabilities are above the threshold. Due to the nature of the volumes to be segmented, one voxel cannot belong to more than one bone structure at the same time. To address both issues, I opted for using a voting system that simply selects the probability with the largest value according to [131]

$$Y(i) = argmax(P_i(j)). \tag{27}$$

Table 4. Dice Score for different test volumes. The best value for each column is shown in boldface.

Case	Vertebral Body		Pelvis		Sternum	
	T=0.5	*argmax*	*T=0.5*	*argmax*	*T=0.5*	*argmax*
P01_d2	0.7938	0.9239	**0.8858**	**0.9323**	0.8049	0.8770
P02_d1	0.8552	0.9060	0.8639	0.9191	**0.8220**	**0.8966**
P02_d3	**0.8745**	0.9190	0.8183	0.9158	0.8218	0.8820
P03_d3	0.8227	**0.9354**	0.8454	0.9145	0.7867	0.8716
P05_d1	0.8096	0.9226	0.8426	0.9033	0.8154	0.8439
P06_d1	0.8436	0.9265	0.8794	0.9128	0.8176	0.8614
P15_d2	0.7963	0.8935	-----	-----	-----	-----
P16_d2	0.8195	0.9029	-----	-----	-----	-----
Average	0.8269	0.9162	0.8559	0.9163	0.8114	0.8721

Table 5. Comparison of the proposed method with competing vertebral body segmentation techniques from the literature.

Work	Score		Method
	Mean Dice	TPR	
This work	0.916	0.915	3D U-Net
Carson (2021) [15]	0.922	-----	Multi-view ensemble U-Net
Nguyen et al. (2016) [117], [118]	-----	0.917	Graph Cut
Yao et al. (2015) [106]	0.936	-----	Geometric Model
Blumfield (2014) [132]	-----	0.963	Statistical Model

Eq. (27) indicates that for a single voxel i, the predicted class is the index of the probability map entry (0 for background, 1 for vertebral body, 2 for pelvis, and 3 for sternum) presenting the largest probability. This eliminates the possibility of duplicates and gives more flexibility for the edge voxels than

using a fixed preset threshold. Table 4 shows a comparison of the Dice score with a fixed threshold and using the argmax voting scheme just described. The last two volumes only contain annotations for the vertebral bodies. There is an improvement of over 0.05 points when using argmax voting. The sternum presents the lower values due to its small size, making it more sensitive to false positive and false negatives values.

Table 5 shows a comparison with other methods for vertebral body segmentation. I purposely did not include works from the Verse challenge, nor many others, since their works process the whole vertebrae and not the vertebral body only. The works from Yao et al. [106] and Blumfield [132], although showing outstanding performance, require a high resolution in the axial plane and have not been evaluated on undersampled volumes like the scans from the HSCT dataset. The Dice score obtained with the model proposed in this thesis is directly comparable to the multi-view ensemble U-Net proposed by Carson [15], which also uses the same HSCT dataset. The benefit that the present work offers relative to [15] is that training requires a smaller number of iterations (approximately 43% of total iterations required by Carson's method).

3.2.2. Qualitative Results

The predicted probabilities were exported to NIfTI format and loaded, along with the original scan and ground-truth mask, to the Slicer software [120] for visualization. Sample scans for each segmented class are shown in Figures 12-14. The value of the predicted probabilities for each class are represented in the middle column with a shade from the color bar located at the bottom of each figure, ranging from dark blue for the lowest values to dark red for the highest ones. The first thing to notice in the predictions located at the

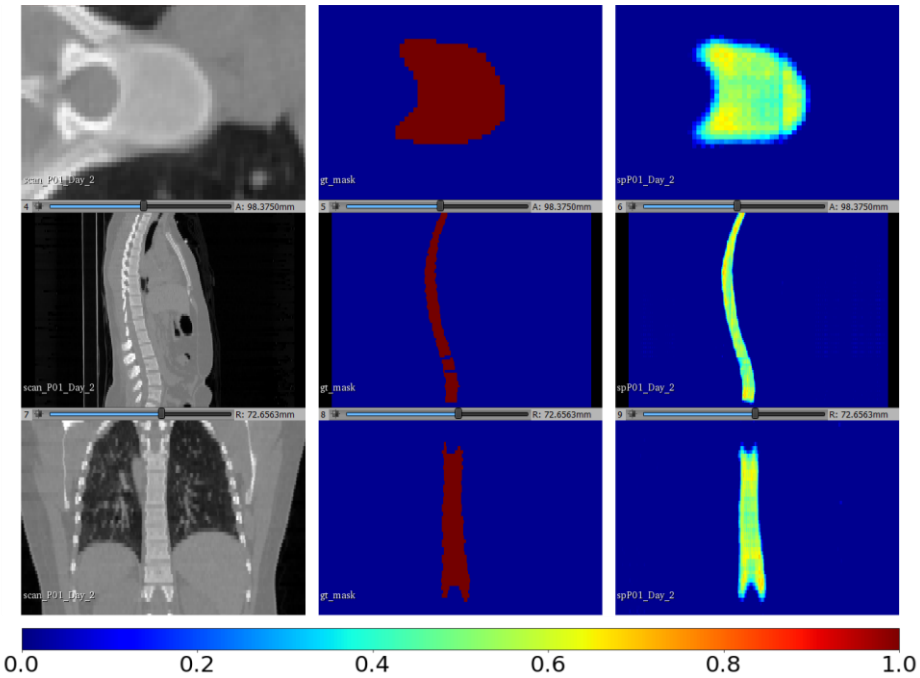


Figure 12. CT (left), vertebral body ground truth mask (middle) and predictions (right) for a test case. The color map shown at the bottom ranges from 0 (dark blue) to 1 (dark red).

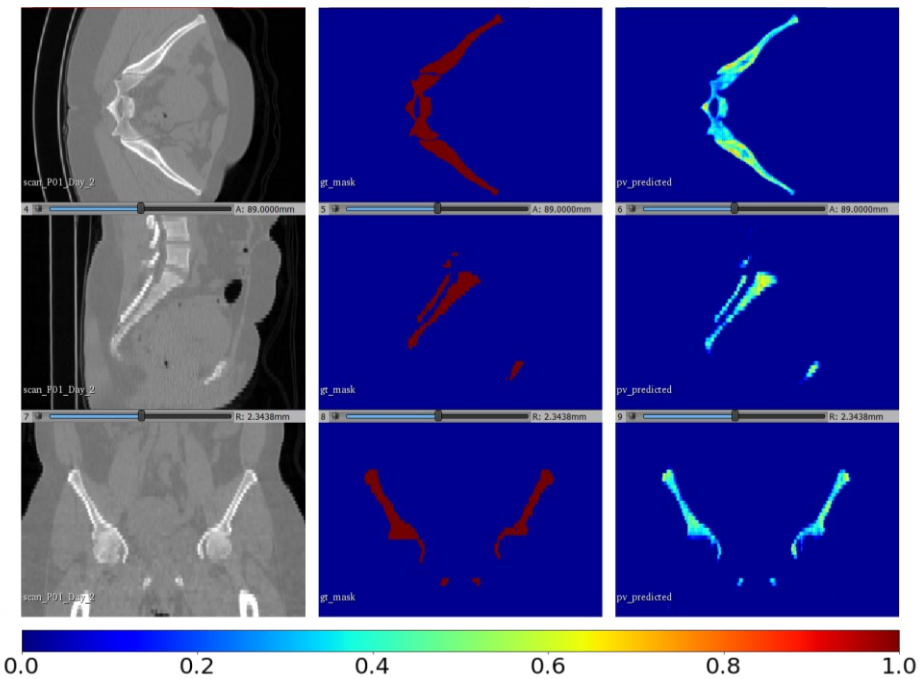


Figure 13. CT (left), pelvis ground truth mask (middle) and predictions (right) for a test case. The color map ranges from 0 (dark blue) to 1 (dark red).

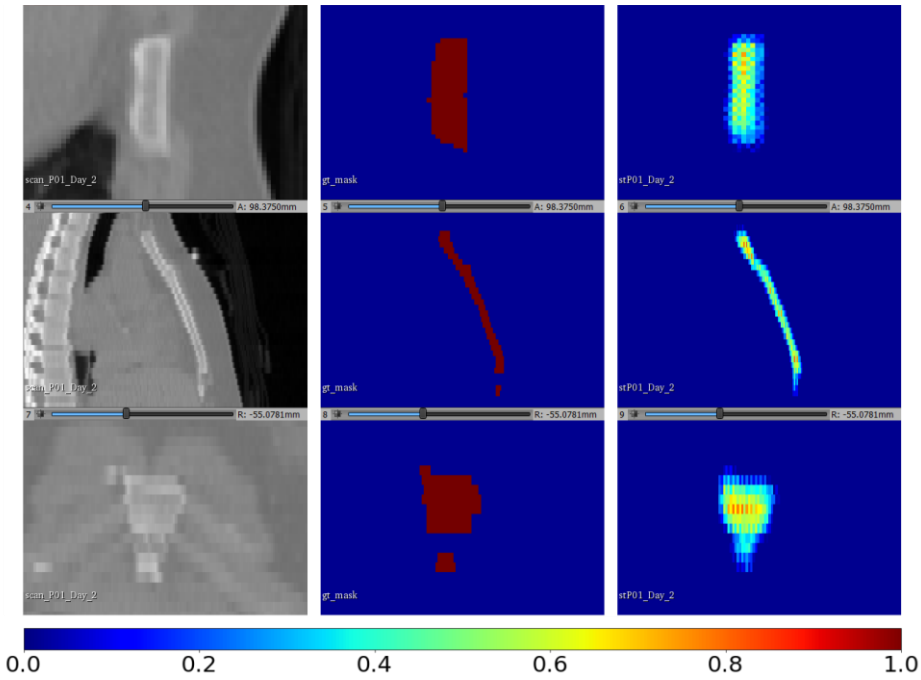


Figure 14. CT (left), sternum ground truth mask (middle) and predictions (right) for a test case. The color map ranges from 0 (dark blue) to 1 (dark red).

rightmost side of Figures 12-14 is the predominance of true negative voxels presenting the lowest probability. The predictions for each class, particularly in the boundary region, are concentrated in the middle range, thus selecting a proper threshold value requires a more sophisticated criteria than simply using the preset of 0.5. This is where argmax comes in handy for prioritizing non-background probabilities. Also, in the rightmost panel of the middle line in Figure 12, the segmented volume for the vertebral bodies appears as a single large object, even though gaps due to the intervertebral discs in the lumbar region are clearly visible in the ground truth mask. A similar issue occurs in Figure 14 in the sternum segmentation with the small gap that is visible in the ground truth mask due to the costal cartilage.

A 3D rendering of the resulting structures after applying the argmax function are shown in Figures 15-17. The true positive voxels, represented in gray, constitute the majority of the segmented volumes. False positive and

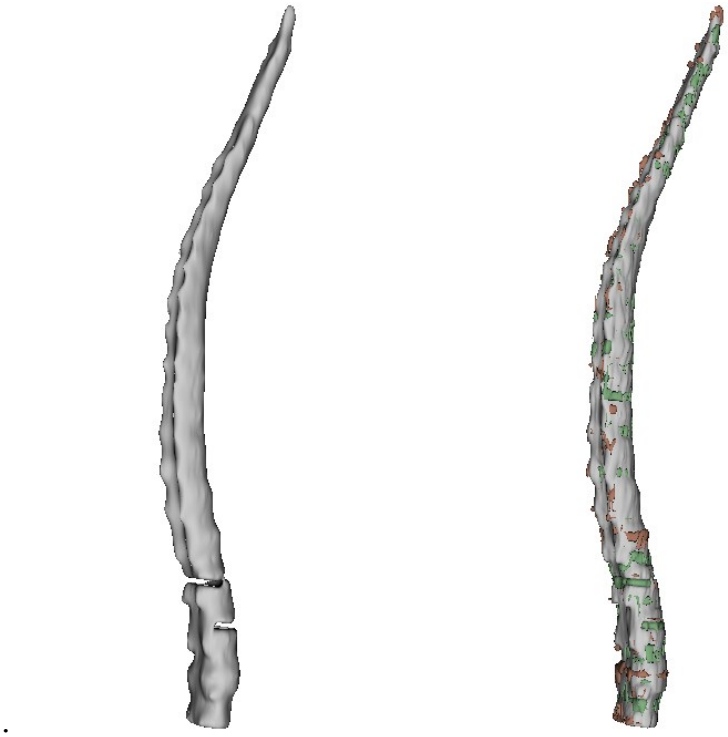


Figure 15. Ground truth (left) and segmentation result (right) for the vertebral bodies on a sample patient. True positives are shown in gray, false negatives are shown in red and false positives are shown in green

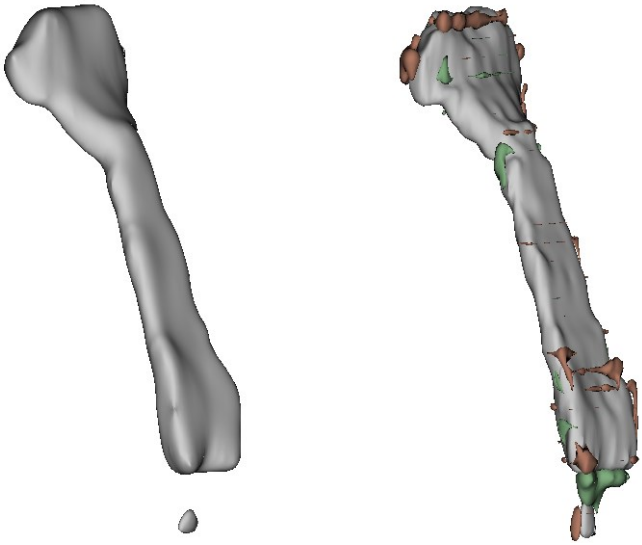


Figure 16. Ground truth (left) and segmentation result (right) for the sternum class on a sample patient. True positives are shown in gray, false negatives are shown in red and false positives are shown in green.

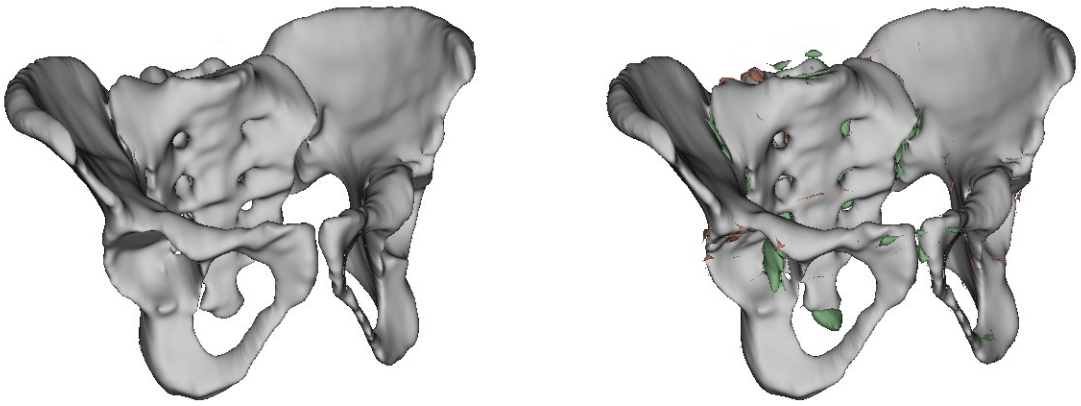


Figure 17. Ground truth (left) and segmentation result (right) for the pelvis class on a sample patient. True positives are shown in gray, false negatives are shown in red and false positives are shown in green.

false negative voxels, represented in green and red, respectively, can be appreciated in the boundaries of the volume. The false negative voxels become more notorious in the gaps from the spine and sternum as previously discussed. These inconsistencies have a negative impact in the Dice score, as indicated by the results from section 3.2.1. Additional segmentation examples are shown in Appendix A.

3.3. Discussion

In this chapter I presented a 3D U-Net model for multiclass segmentation of three bone structures: vertebral body, pelvis, and sternum. Due to the lack of annotations for the three classes and the imbalance with respect to the background voxels, data augmentation played a major role during the training process. An improvement of approximately 0.09 points in the mean Dice score was obtained when using argmax instead of a threshold when generating predicted masks for vertebral bodies, resulting in a value of 0.916.

By using transfer learning, it would be possible to enhance the model's performance or to add new segmentation classes such as femur and liver.

However, as with the three segmentation classes demonstrated in the examples given here, performance on any additional added classes would be limited by the availability of annotated ground truth data.

Chapter 4. Instance Segmentation of Vertebral Bodies

After bone segmentation based on the CT scan data, the next step in the proposed framework for SUV extraction is to identify each one of the individual vertebrae from the segmentation mask generated by the 3D U-Net model. Proper vertebrae identification is still a challenge due to the geometric irregularities in their anatomy. I attempted to use the techniques described in [115] and [133] without success, since those methods require a higher axial resolution than the one present on the HSCT dataset. Motivated by the approach reported in [106], I designed an algorithm for instance segmentation of vertebral bodies using anatomical characteristics of the vertebrae.

4.1. Anatomical Priors

The low axial resolution of the HSCT dataset adds an extra level of difficulty to vertebrae identification due to the CT axial slice thickness being on the same order as or even thicker than the thickness of intervertebral discs of the cervical region [11]. However, there are some anatomical characteristics that can be exploited for vertebrae identification. By examining the difference of intensities in a single vertebra, as indicated in Figure 18, the authors in [106] developed a technique for identifying the sections of a vertebra. Motivated by the fact that the HUs for the soft bone are usually lower than the hard bone from the pedicles, [106] identified some landmarks within the vertebrae that were used for generating cutting planes to extract four regions: vertebral body, spinous process and left and right transverse processes. The approach that I took was essentially the inverse of the one described in [106]: by using the vertebral mask obtained from segmentation, my goal was to identify the whole vertebra. Specifically, my proposed solution is to identify the

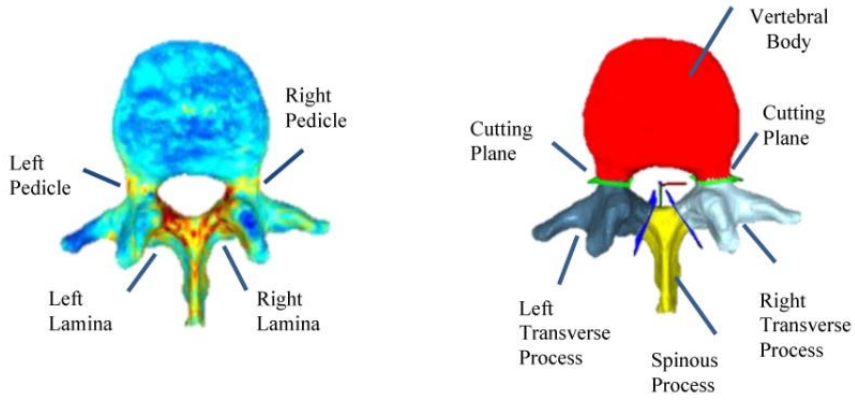


Figure 18. Partitioning of a vertebra. Left: heat map of typical intensity values on the vertebra. Right: Four sections obtained from a vertebra. Extracted from [106].

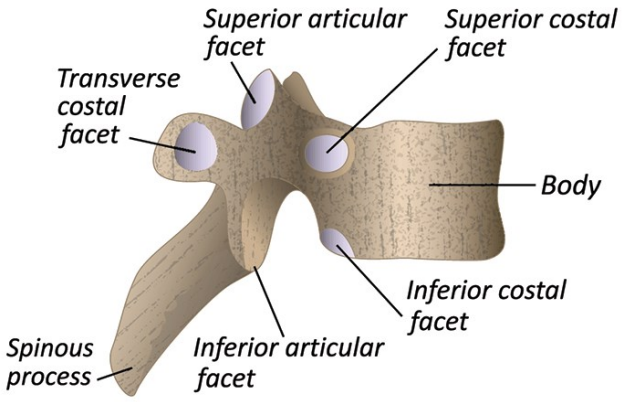


Figure 19. Side view of a vertebrae. Extracted from [134].

corresponding transverse processes for each vertebral body and use them to identify the boundaries between the individual vertebrae.

I leveraged the fact that transverse processes are axially aligned near to the top of the vertebral body, as shown in Figure 19, to aid in identification of the individual vertebral bodies in the HSCT dataset CT scans. Another relevant prior that I used in developing the algorithm is the fact that the contour of the spine loosely resembles the shape of a double "S" when viewed sagittally. The inward and outward curvatures are named according to the

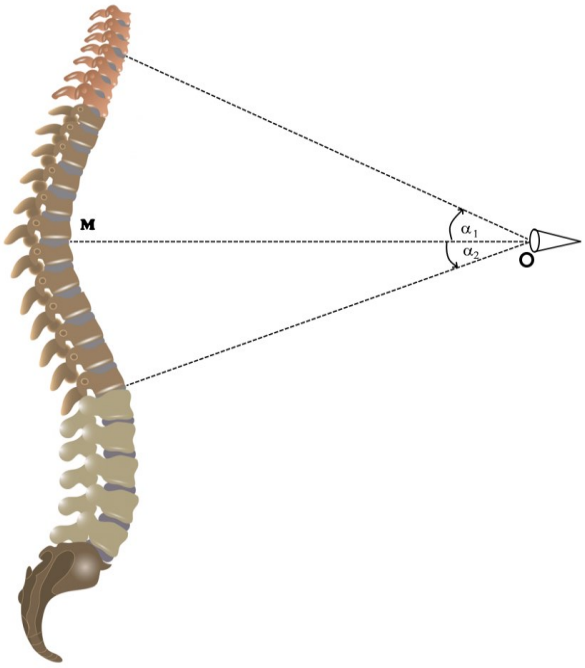


Figure 20. Examples of the angles of vision generated by the observer located at point O, when looking at the intervertebral discs from a sagittal view of the spine. Spine model extracted from [134].

region where they are located, viz. cervical, thoracic, lumbar and sacral curvatures [27]. Now consider a sagittal view from the spine and the point M located at the middle of the thoracic curvature (also known as the thoracic kyphosis), as shown in Figure 20. An observer located at point O to the right of the figure and at the same height as the point M will generate positive angles of vision when looking at the intervertebral discs located above the middle point M. For the remaining thoracic vertebrae, the angle generated is negative. A similar observation can be made for the lumbar vertebrae by placing the observer to the left of the figure.

4.2. Methods

The proposed solution consists of extending the dimensions of the segmented vertebral volume to capture the additional vertebral structures. With the new extended mask, the boundaries between the individual vertebrae

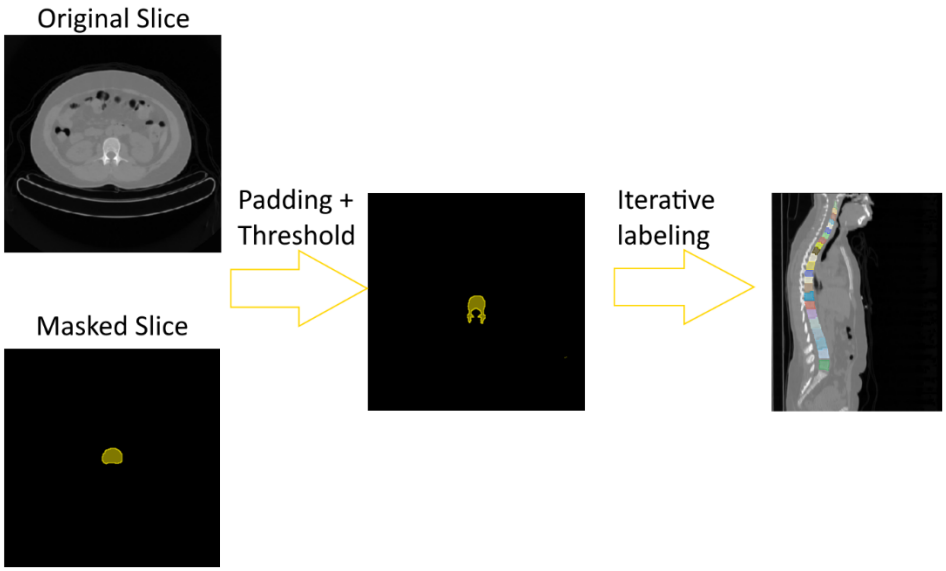


Figure 21. Overview of the proposed method for vertebrae identification.

can be estimated by locating the transverse processes. This technique assigns a unique label to each individual vertebra. Figure 21 shows an overview of the proposed algorithm. For this implementation, and based on the DICOM standard [39], I am considering an LPS+ (left, posterior, superior) anatomical system for the PET/CT scans, meaning that axial slices are parallel to the XY plane while sagittal slices are parallel to the YZ plane. The z-axis and its corresponding *k*-index in the image space present the lowest values at the patient's soles and increase towards the head. The algorithm implementation is detailed below:

Step 1: ROI Extraction: from the segmented volume obtained from the 3D U-Net, extract the region of interest by selecting the voxel coordinates containing the spinal column segmentation.

Step 2: Mask Padding: for each axial slice, get the start and end coordinates containing the original mask for label 1 (vertebral body). Then, starting from the lower-left corner of this vertebral body mask, create an additional bounding

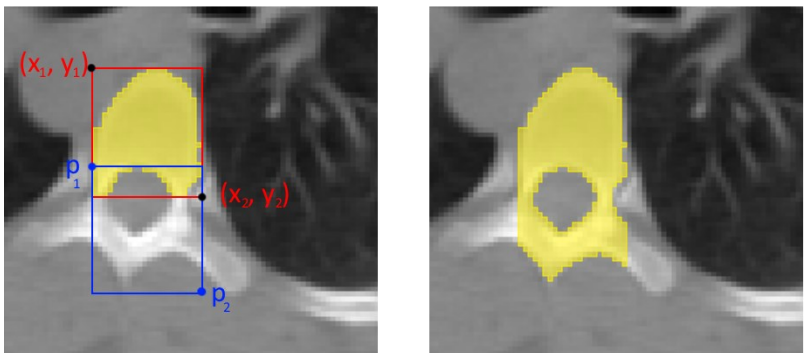


Figure 22. Left: original vertebral body mask (yellow), bounding box of original vertebral body mask (red) and new bounding box for capturing the pedicles and portions of the transverse processes (blue). Right: Resulting augmented mask.

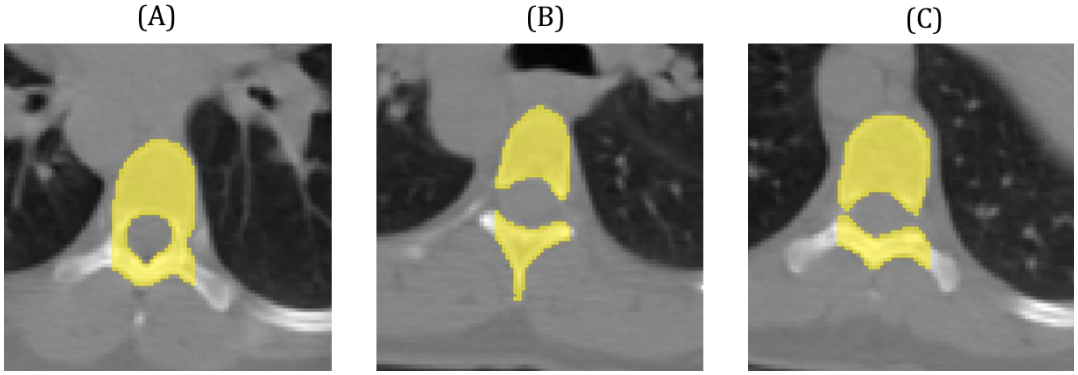


Figure 23. Three possible detections using the extended mask. (A) Vertebral body with pedicles. (B) Spinous process. (C). Transverse process without visible pedicle.

box to capture the transverse process. Considering an original vertebral body mask with start coordinates (x_1, y_1, z) and end coordinates (x_2, y_2, z), with z representing the current slice, the new bounding box coordinates, as shown in blue in Figure 22, are given by

$$p_1 = (x_1, y_2 - \Delta y_1, z), \tag{28}$$

$$p_2 = (x_2, y_2 + \Delta y_2, z), \tag{29}$$

where Δy_1 and Δy_2 are the padding values. Using the new bounding box, apply a threshold to the original CT scan for selecting additional bone structures

from the vertebra while discarding the soft tissue voxels which are characterized by a much lower HU value. Finally, concatenate the newly selected voxels with the original mask. The idea is to capture the small pedicles that act as a bridge between the vertebral body and the transverse processes. Padding in the horizontal direction was not considered in order to prevent capturing additional unwanted bone structures like the ribs in the thoracic region.

Step 3: Connected component analysis: starting from the topmost slice, corresponding to a cervical vertebra, perform a 2D connected component analysis on the current slice, with an 8-connectivity. Filter out the small regions that may appear after thresholding and find the enclosed region R that includes the original mask M. Compare the bounding boxes of R and M. If R is considerably larger, this indicates the presence of a pedicle. Otherwise, the isolated islands may belong to the spinous process or to the superior articular facet. Examples of the possible outcomes are shown in Figure 23.

A class map is then generated by examining the obtained regions. In the absence of pedicles, assign the same label as the previous slice. When a pedicle is detected and the previous slice did not include a pedicle, assign a new label (this indicates that the current axial slice contains the start of a new vertebra). Otherwise, a new vertebra is not detected and the voxels contained in the vertebral body mask of the current axial slice should be labeled the same as the label of the current vertebra.

Step 4: Boundary refinement: For the cases where a change of label occurs between two slices, define the parametric line between the transitional slices on the sagittal plane:

$$(y, z) = (y_0, z_0) + t(m_y, m_z). \tag{30}$$

The values for the starting point (y_0, z_0) and the slopes (m_y, m_z) in (30) will depend on the curvature of the spine in a sagittal view, as indicated by Figure 20. Using the mask centroids for the current k-th slice, I approximate the curvature of the spine by

$$\frac{\Delta z}{\Delta y} = \frac{k+l-k}{y_{c_{k+l}} - y_{c_k}} = \frac{l}{y_{c_{k+l}} - y_{c_k}}, \tag{31}$$

where l is an integer representing the index-based distance from an axial slice to the current reference slice k, and $y_{c_k}, y_{c_{k+l}}$ are the vertical coordinates of the mask centroids in slices k and $k + l$ respectively.

Considering a positive value for l, obtaining a negative value in (31) indicates that the current slice is above the midpoint of the thoracic curvature. Eq. (30) may then be expressed in matrix form according to

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} p_{2,y_k} \\ k \end{bmatrix} + t \begin{bmatrix} p_{2,y_k} - p_{1,y_{k-l}} \\ k \quad -(k-l) \end{bmatrix} = \begin{bmatrix} p_{2,y_k} \\ k \end{bmatrix} + t \begin{bmatrix} p_{2,y_k} - p_{1,y_{k-l}} \\ l \end{bmatrix}, \tag{32}$$

where p_{1,y_k} and p_{2,y_k} denote the start and end y-coordinates of the bounding box for the k-th slice. The $(k-l)$-th index corresponds to an axial slice located towards the lumbar region with respect to the current slice. The obtained expression from (32) is then extended through the X-plane, dividing the YZ plane in two semi-planes. The voxel labels are reassigned so that the voxels located within the k-th and $(k-l)$-th slice and belonging to the superior semi-plane generated by the parametric line from (30) are assigned the same label as the vertebral body identified in the axial slice k. The label from the axial slice $k-l$ is assigned to the voxels located in the inferior semi-plane. This situation is illustrated in Figure 24, where the boundary between C6 and C7, defined by the parametric line l_1, was calculated using (32).

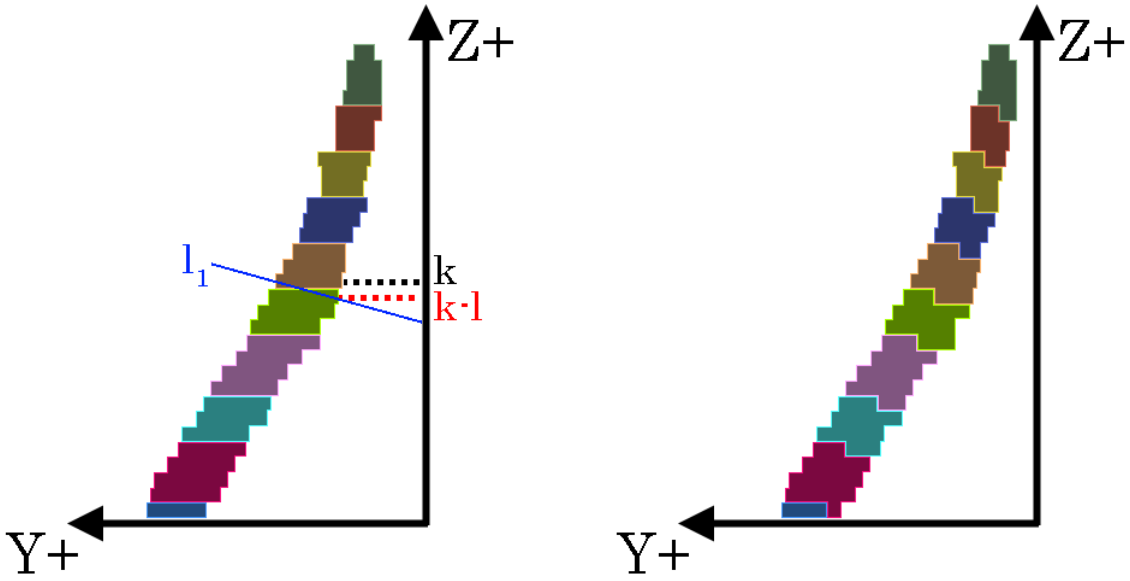


Figure 24. Left: Sagittal view of the class map for the individual vertebrae (from top to bottom: C2-C7, T1-T4). The parametric line l_1 passes through the bottom-left corner of C6 and the top-right corner of C7, dividing the YZ plane into two semi-planes. Voxels belonging to C7 and located in the superior semi-plane generated by l_1 and within *k* and *k-l* are reassigned to C6. Right: resulting class map after applying the boundary refinement to the case presented in the left side.

The situation is similar when the approximated curvature obtained in (31) is positive. In that case, the equation for the parametric line (30) is given by

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} p_{1,y_k} \\ k \end{bmatrix} + t \begin{bmatrix} p_{1,y_k} - p_{2,y_{k-l}} \\ k \quad -(k-l) \end{bmatrix} = \begin{bmatrix} p_{1,y_k} \\ k \end{bmatrix} + t \begin{bmatrix} p_{1,y_k} - p_{2,y_{k-l}} \\ l \end{bmatrix} \quad (33)$$

and the labels are reassigned as previously discussed for Eq. (32), by extending the parametric line obtained in (33) through the X-plane.

Step 5: Mask refinement: As discussed in Chapter 3, the resulting segmented vertebral body also includes the vertebral discs from the lumbar region. For this particular case, the contrast between slices is higher due to the larger dimensions of both the vertebra and discs located in this region. A localized threshold then improves the label identification for lumbar vertebrae L1-L5.

Table 6. Dice scores obtained on the test scans, grouped by vertebral region.

Scan	Dice score by vertebral region		
	Cervical	Thoracic	Lumbar
Scan 1	0.903	0.837	0.881
Scan 2	**0.911**	0.829	**0.899**
Scan 3	0.896	0.825	0.874
Scan 4	0.905	**0.901**	0.891
Scan 5	0.876	0.853	0.88
Scan 6	0.890	0.812	0.861
Scan 7	0.904	0.810	0.882
Average	0.898	0.838	0.882

4.3. Results

4.3.1. Quantitative results

For the vertebral body segmentation algorithm I have proposed in this chapter, calculating standard segmentation metrics on the HSCT dataset is not possible due to the lack of annotated ground truth for the individual vertebral bodies. Instead, I selected full-body scans from the Verse dataset [112]-[114] and resampled both the ground truth and annotation volumes to match the voxel dimensions from the HSCT dataset. Since the annotations for the VerSe dataset include the whole vertebra, I chose to remove the additional vertebral parts to generate new ground truth volumes containing annotations for only the vertebral bodies. Finally, I ran the instance segmentation algorithm with the selected test cases and proceeded to calculate the Dice scores for each vertebrae class. C1 was not considered on the calculations due to the peculiarities of its anatomy. The results, grouped by region, are presented in Table 6. The vertebrae from the cervical region show the best results. For these, the boundary refinement works best due to their small size

Table 7. Max. Dice scores obtained in [106] for segmentation of vertebral bodies.

Method	Max. Dice per region	
	Thoracic	Lumbar
Method 1 [107]	0.92	0.94
Method 2 [108]	0.88	0.86
Method 3 [109]	0.96	0.97
Method 4 [110]	0.96	0.97
Method 5 [111]	-----	0.96

and absence of other nearby bone structures. The thoracic vertebrae show the most variability, most likely due to the presence of rib structures that are erroneously classified as vertebral body when using the extended mask. Although the results from the lumbar region seem better than the thoracic case, the boundary approximation introduces error due to the larger size of the vertebrae in this region.

The authors from [106] evaluated five methods [107]-[111] submitted to the "2nd MICCAI challenge for spine segmentation" of 2014 for segmentation of the whole vertebrae and then they proceeded to segment the vertebral bodies by identifying landmarks based on HU intensity on the dataset provided in [135], which presents CT scans with a slice thickness ranging from 0.7 mm to 2 mm. Table 7 shows a summary of the maximum Dice score obtained for the vertebral body segmentation on each evaluated method. The first thing to notice is that instance segmentation for the cervical vertebrae is not performed by any of the methods, which represents an advantage for my proposed method. Method 5 [111] only works on the lumbar region. By inspecting the values from Table 6, it can be observed that the Dice scores obtained by my proposed algorithm on the generated undersampled volumes are roughly

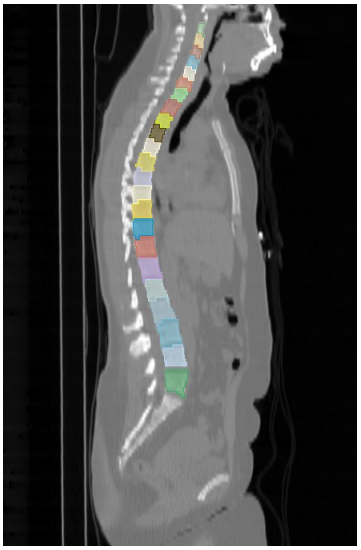


Figure 25. Sagittal view of the class map obtained for the vertebrae.

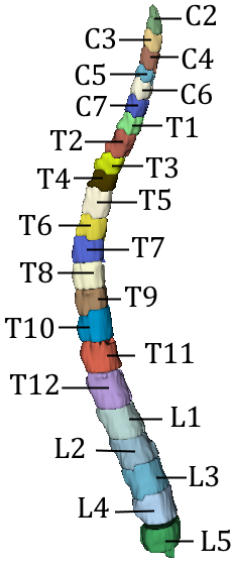


Figure 26. 3D rendering of vertebral bodies labeling.

similar to the results obtained for Method 2 [108] on the high-axial resolution scans from the MICCAI dataset [135].

4.3.2. Qualitative results

I ran the proposed algorithm against scans from the HSCT dataset for 22 patients corresponding to the 28th day post-HSCT. For all the tested volumes,

the 23 desired vertebrae were identified. The major discrepancies occurred at the boundaries between two vertebrae. In the HSCT dataset, the intervertebral discs are almost indistinguishable to the naked eye in the cervical region and become more detectable while traversing the spine. In the thoracic and lumbar regions, the differences generated by the approximation method are more visible. Figure 25 shows a sagittal slice from a sample CT scan with the vertebral class map overlapped, and Figure 26 shows a 3D rendering for the same CT scan. Each color represents a unique vertebral body, starting with C2 at the top on both figures, followed by C3 through C7, the thoracic vertebrae T1-T12 and finally the lumbar vertebrae L1-L5, with L5 located at the bottom. In both figures, the boundary between two vertebrae is delimitated by the parametric lines resulting from Eq. (30), as discussed in section 4.2. Additional examples of individual vertebral body segmentation are shown in Appendix B.

4.4. SUV Measurement on Vertebral Bodies

One of the major motivations for identifying and detecting the individual vertebral bodies in this project is to obtain automated SUV measurements from the PET scans. As I discussed in Chapter 3, the patients scans from the HSCT study were acquired using the joint PET/CT modality. However, this is a two-stage process: first the patient is scanned with low-dose CT, and then the PET scanning takes place. The scan resolutions of the two modalities are different. On the CT scan, the voxels are anisotropic with an approximate size of 1.17mm × 1.17mm × 5mm. On the other hand, the PET voxels are isotropic with size 4mm × 4mm × 4mm. Examples of sagittal slices for the PET and CT scans are shown in Figure 27. The difference in resolution on both imaging techniques presents an issue for acquiring the desired SUV measurements. Specifically, the data required for calculating SUV is located on the PET scans. However, the PET scans lack sufficient resolution for detecting the marrow cavities of the bones, especially when the metabolic activity measured by the

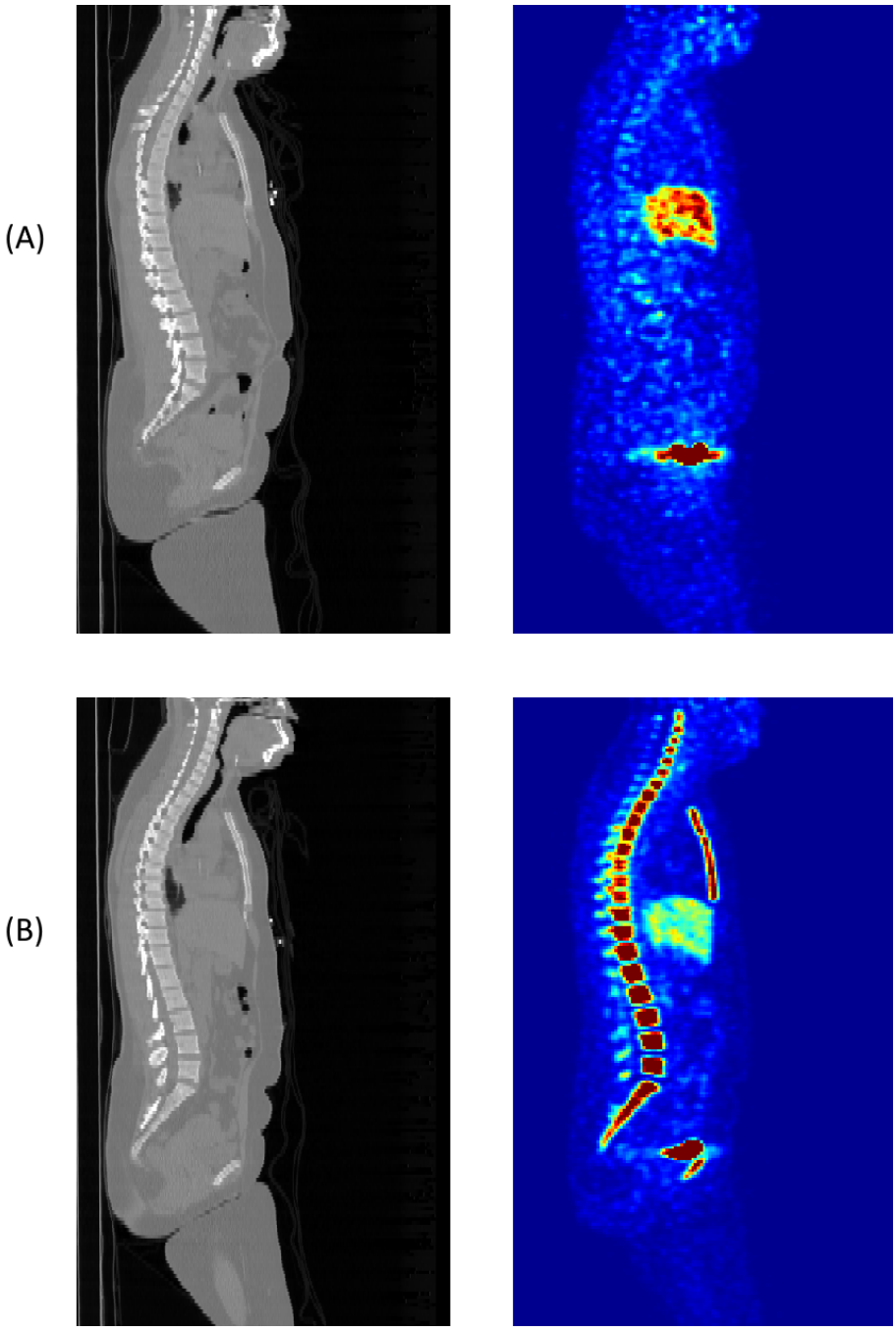


Figure 27. CT (left) and PET (right) sagittal slices from a sample subject. *A:* Scanning one day before HSCT treatment. *B:* 28 days after treatment.

radiotracer is low as it typically is on the scans from the day before the HSC transplant and often is even on scans acquired three to five days after

transplant, as shown in the top-right slice from Figure 27. Additionally, the initial reference point varies when changing the modalities, causing a misalignment between the CT and PET images. In order to obtain the desired SUV measurements, the approach I took is to extract the desired bone structures from the CT image space and then translate them to the PET image space. I implemented this process using the spatial information contained in the affine matrices obtained from the scans. Using the acquired segmentation mask for individual vertebral bodies, I converted the whole image (i.e., the joint CT and PET scans) to the PET space by using the expression from Eq. (8) and with aid of the Scikit library [124], resulting in a new mask conforming to the voxel coordinate system of the PET scans. Some padding and cropping were necessary to match the exact volume dimensions. After this conversion, the information from each individual segmentation class can then be accessed by simply filtering the desired label (1 to 23, starting from C2). The values of the obtained PET voxels were then stored in a list for further analysis to obtain the desired SUV measurements. Upsampling techniques such as interpolation or creating an intermediate image space for both the PET and CT images were not considered since doing so would be likely to distort the precalculated SUV values that are tightly coupled to the PET voxel dimensions.

4.4.1. Results

I ran the instance segmentation algorithm on the scans corresponding to the 28th day after transplant for 22 patients to identify 23 unique vertebrae: C2-C7, T1-T12 and L1-L5. Then, by matching the obtained mask to the voxel coordinate system of the PET images, I extracted the SUV values for each one of the individual vertebrae to calculate some statistics of relevance for clinical study, including the mean, median, maximum, and standard deviation of the SUV. The same procedure was performed for the sternum and pelvis masks obtained from the 3D U-Net. An example of the SUV distribution inside the

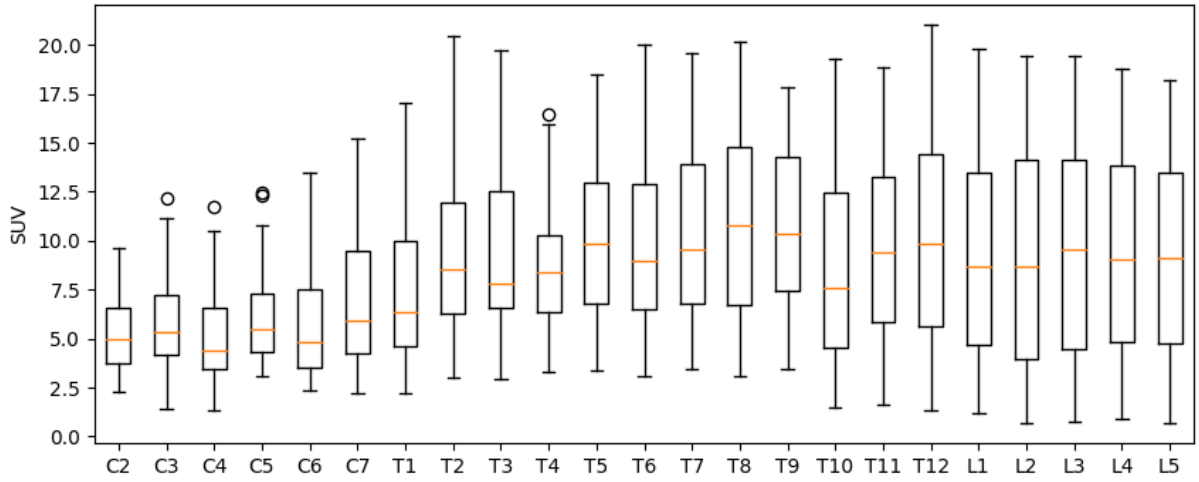


Figure 28. SUV distribution of the vertebral bodies from a sample patient, on the 28th day after HSCT treatment. The boxes extend from the first quartile to the third quartile. Orange segments inside each box indicate the median value. Whiskers extend to the minimum/maximum value within ± 1.5 times the interquartile range. The circles represent values outside that range.

vertebral bodies of a sample patient is shown in the boxplots from Figure 28. The boxes in this figure extend from the first quartile Q_1 to the third quartile Q_3 of the SUV extracted from each vertebra. Orange segments inside each box indicate the median value. The whiskers extend to the minimum and maximum value within ± 1.5 times the interquartile range $Q_3 - Q_1$. The small circles represent outliers, i.e., values that fall outside said range. Additional examples of the SUV distribution are shown in Appendix C.

For comparison, I took the SUV results obtained by Carson [15] on the same subset of patients from the HSCT dataset. I calculated the difference between the two methods on the four statistics (mean, median, maximum, standard deviation) of each individual vertebra. Figure 29 shows the distribution of the magnitudes of the differences for a sample patient in the form of boxplots. As previously discussed, the boxes extend from the first quartile Q_1 to the third quartile Q_3, the whiskers extend to ±1.5 times the interquartile range, and outliers are represented by the small circles. The best agreement in the data

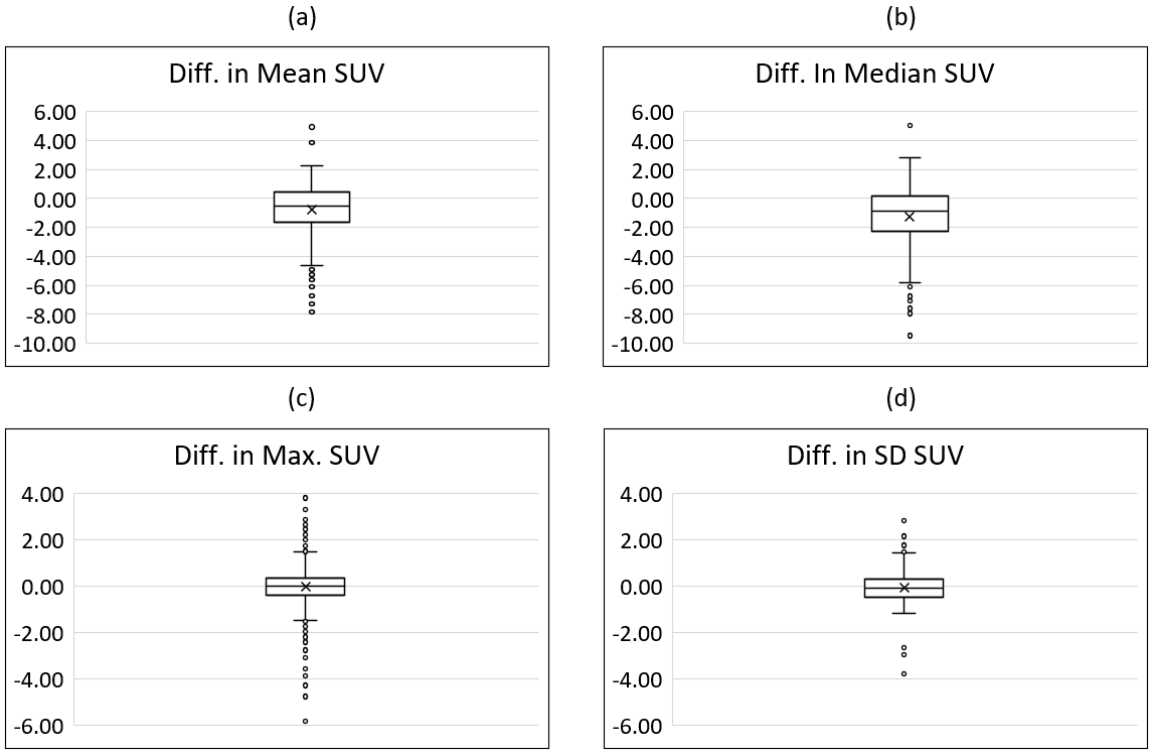


Figure 29. Distribution of the magnitude difference in the statistics calculated in the present work and [15]. a: Mean b: Median c: Maximum d: Standard Deviation. The boxes extend from the first quartile to the third quartile. Whiskers extend to the minimum/maximum value within ± 1.5 times the interquartile range. The circles represent values outside that range. The large segment inside the box indicates the median value and the "X" marker the mean.

distribution occurs between the max SUV and the standard deviation (SD) of the SUV, as indicated by the interquartile ranges located within ±2 SUV. This is an encouraging result, since I noticed that the peak SUV values are usually located towards the center of the vertebral body, thus indicating a proper identification of the vertebra centroid by both methods. The method described by Carson in [15] for vertebrae identification, assigns a single label to each axial slice, i.e., it assumes that the boundary between two vertebrae is flat, unlike my proposed algorithm that considers a slope between the vertebrae, as discussed in section 4.2. This translates into a more prominent variability in

the distribution of mean and median, as indicated by the two boxplots located at the top of Figure 29.

4.5. Discussion

In this chapter I presented a method for instance segmentation of individual vertebral bodies, making use of some anatomical properties of the vertebrae for estimating the boundaries between vertebrae. This method, unlike others evaluated, was designed considering undersampled CT scans like the ones in the HSCT dataset. For volumes with better axial resolution a refinement of the boundary detection is needed, as indicated by the results obtained in the lumbar region which usually show higher contrast. The segmentation masks obtained from the CT scans were then translated to the PET image space for extracting the SUV from each vertebra and computing statistics that may be used as indicators of the patient recovery after treatment [7], [8]. The PET image space was chosen to prevent alterations in the precalculated SUV values; however, if the original PET data is available, other registration methods could be considered for resampling the data and calculating the SUV on the final step. I compared my method with Carson's [15], and there is a strong agreement in the maximum values obtained from both results as seen in Figure 29. Carson's technique wisely exploits the information from the dual PET/CT modality to extract SUV values. However, this requires that the radiotracer exhibits high metabolic activity in the patient, which is unlikely to be observed in the early stages of the HSCT treatment regimen when the immune system is ablated. With my proposed solution, the SUV can be extracted from scans at any time, which could be used by medical professionals for monitoring the condition of the patient throughout the transplant and recovery process.

Chapter 5. CNN Classifier for post-HSCT Evaluation

In the previous chapter I presented an algorithm for extracting and calculating SUV statistics (mean, standard deviation, max, and median SUV) from the segmentation masks obtained in Chapter 3. Although obtaining the SUV from the PET/CT scans is useful for quantifying the metabolic activity of the bone marrow, its application and interpretations are still subject of debate in the medical community, generating divided opinions [136]-[139]. For the particular case of the HSCT study by Williams et al. [7], [8], the purpose of ablating the bone marrows is to eliminate the carcinogenic cells. Stem cell transplantation (HSCT) can only be performed after the patient is certified to be cancer free. Once the transplant has occurred, hematopoietic activity can be sensed and evaluated by the PET imaging. Dr. Holter (2021) indicated that high SUV measurements do not necessarily imply an optimal patient recovery [12], since the high activity may be caused by either cancerous cells reproliferating or by normal hematopoietic activity. The spatial distribution of the SUV within the marrow cavities of the bones may be a better indicator of successful engraftment of the patient. Certain patterns of metabolic activity that are believed to be potentially indicative of normal recovery versus graft failure versus relapse have been repeatedly observed by the physicians in the HSCT study as illustrated by the two examples shown in Figure 30. What is believed is that normal engraftment is potentially indicated by a semi-concentric pattern of the detected cell activity about a central location in the marrow cavity, as illustrated in the left image of Figure 30. Comparatively irregular patterns of activity, visually resembling “sacks” inside the marrow cavity as illustrated in the right image of Figure 30, are believed to be potentially indicative of or even predictive of relapse. If an analysis of the SUV

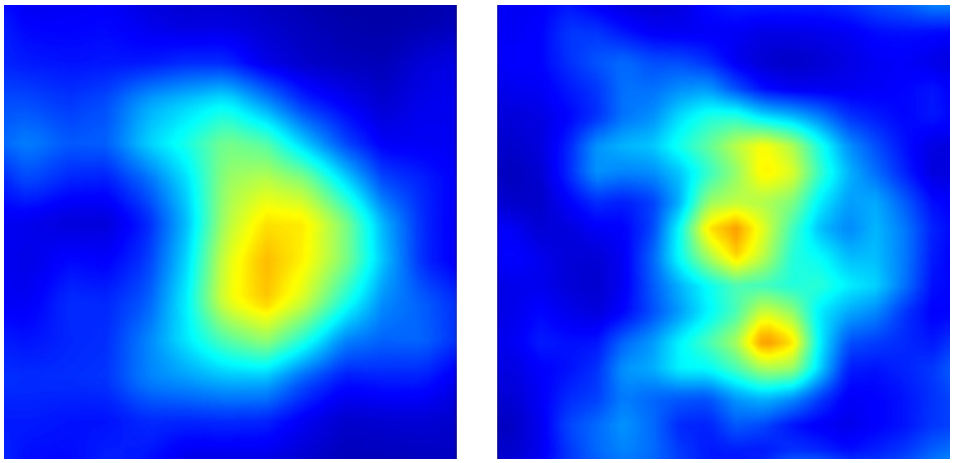


Figure 30. Axial PET slices on the 28th day after treatment. Left: normal engraftment Right: possible relapse. Images have been smoothed for better visualization.

distribution within individual bones could be shown to correlate with relapse, it could lead to a clinically significant means of early relapse prediction, thereby enabling the timely application of life saving therapy modulations that would not otherwise be possible.

With that information in hand, I propose a CNN-based classifier for automatically detecting possible relapsing cases, which could potentially assist medical personnel by reducing the time-consuming task of manually examining numerous individual marrow cavities in multiple scans.

5.1. Implementation Details

5.1.1. Dataset

Using the PET/CT scans from the 28th day post-HSCT, I isolated the individual vertebral bodies with the method described in Chapter 4. I then proceeded to label the thoracic and lumbar vertebrae using the criteria previously described. The cervical vertebrae were not considered due to their small size. Vertebrae presenting a concentric patterns of SUV data as illustrated in the left image of Figure 30 were labeled with a "0", while

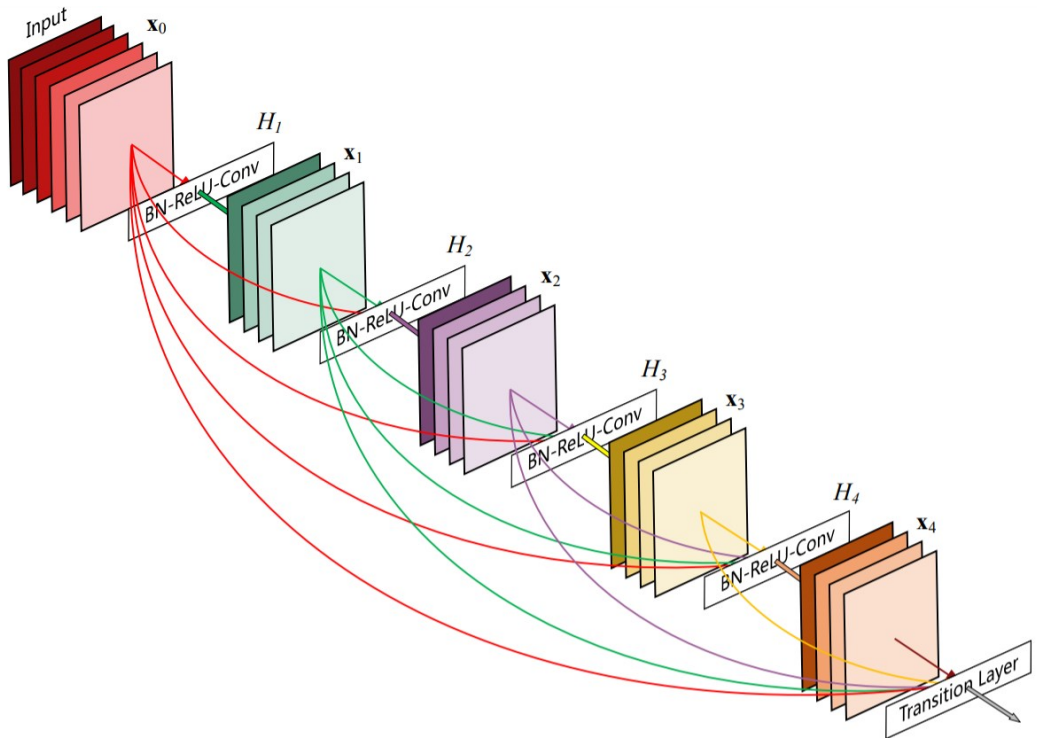


Figure 31. DenseNet architecture, extracted from [16].

irregular SUV patterns such as the one in the right image of Figure 30 were labeled with a "1" to indicate possible relapse. Based on a manual examination of the data, it was determined that six of the selected 22 patients in the HSCT dataset presented the abnormal pattern on at least one vertebral body, coinciding with 83% of the relapsing cases reported by Dr. Holter [12].

5.1.2. Model Architecture

The model I selected for this classification task was based on the DenseNet architecture [16]. As shown in Figure 31, this architecture consists of several densely connected blocks, which are formed by a sequence of batch normalization, activation, and convolutional layers. Specifically, the variant with four densely connected blocks was used, since the successive downsampling that occurs after each stage would be impractical with a larger

number of blocks, due to the small dimension of the segmented vertebral bodies.

Given that I used this model for a binary classification task ("normal" and "irregular" SUV patterns presented in Figure 30), a channel size of 1 was selected for the input and output layers. The dimensions of the input volumes were set to $24 \times 24 \times 24$ voxels, since this is the minimum size that is a multiple of 4 and also contains a whole vertebral body. That constraint on the size was required for supporting the successive downsampling that occurs between layers.

The selected kernel size dimensions were $3 \times 3 \times 3$ for convolution and $2 \times 2 \times 2$ for downsampling. As for the U-Net model, batch normalization was performed after each block. The selected activation function was ReLU [68], [69] since the architecture for the classifier was simpler than the one presented in Chapter 3.

5.1.3. Training

The data was split into three subsets: training, validation, and test. 70% of the available segmented vertebral bodies were used for training, 20% for validation and 10% for testing. Data augmentation was performed on the training data, except that this time only random rotation and flipping was applied to prevent altering the SUV distribution. The ADAM algorithm [70] was again selected as the optimizer function, with a learning rate α equal to 10^{-5}, and exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for the estimates of the first and second moments of the gradient. The reason for changing the selected value for the learning rate with respect to the previous value used (10^{-4}) in the 3D U-Net presented in Chapter 3 was that convergence of the error function during the DenseNet training can be achieved faster than training more complex networks [130]. The error function selected for training was the

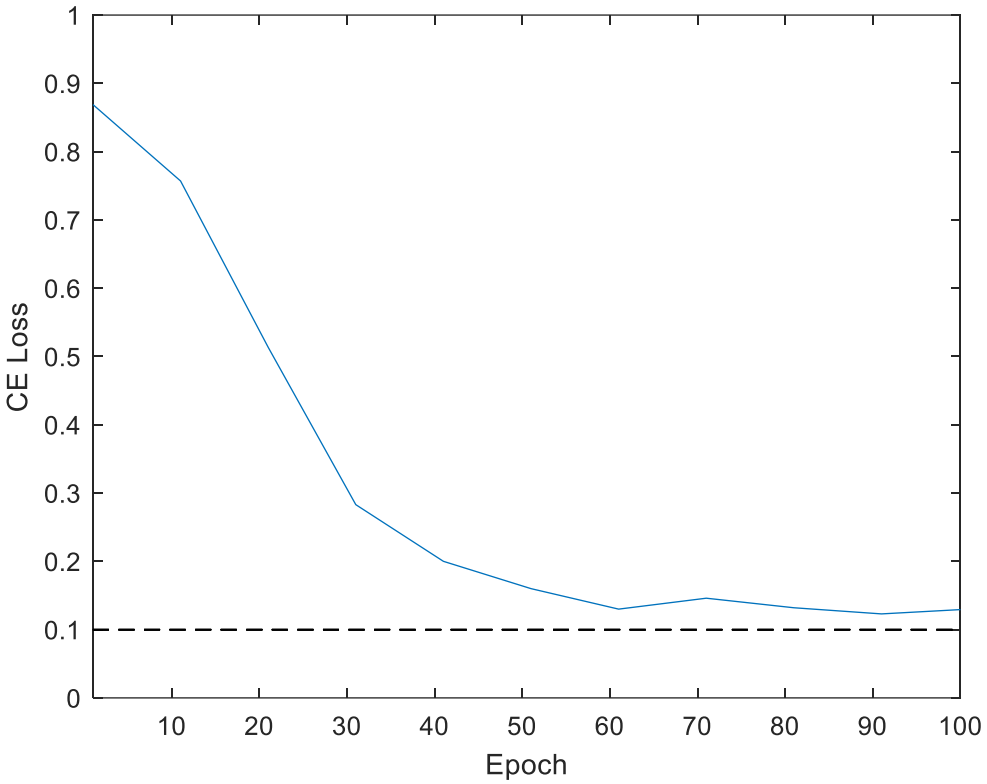


Figure 32 Cross-Entropy loss during training.

cross-entropy loss [56], defined for a binary classifier as

$$CE = -\sum_{i}^{N} g_i \, log(p_i) + (1 - g_i) log\,(1 - p_i), \tag{34}$$

where g_i represents the ground truth values (0 or 1), and p_i the probabilities for the positive class.

The model training ran over 100 epochs, with a batch size of 2. Convergence started at about the 60th epoch, as indicated by Figure 32. An initial run of 200 epochs caused a divergence after the 120th epoch, which was the reason for using a lower epoch size. Like the 3D U-Net case, a margin of approximately 0.1 in the loss function is appropriate to prevent overfitting.

Table 8. Confusion matrix obtained by the DenseNet classifier on the test data.

	Actual Positive	Actual Negative
Predicted Positive	11	4
Predicted Negative	1	23

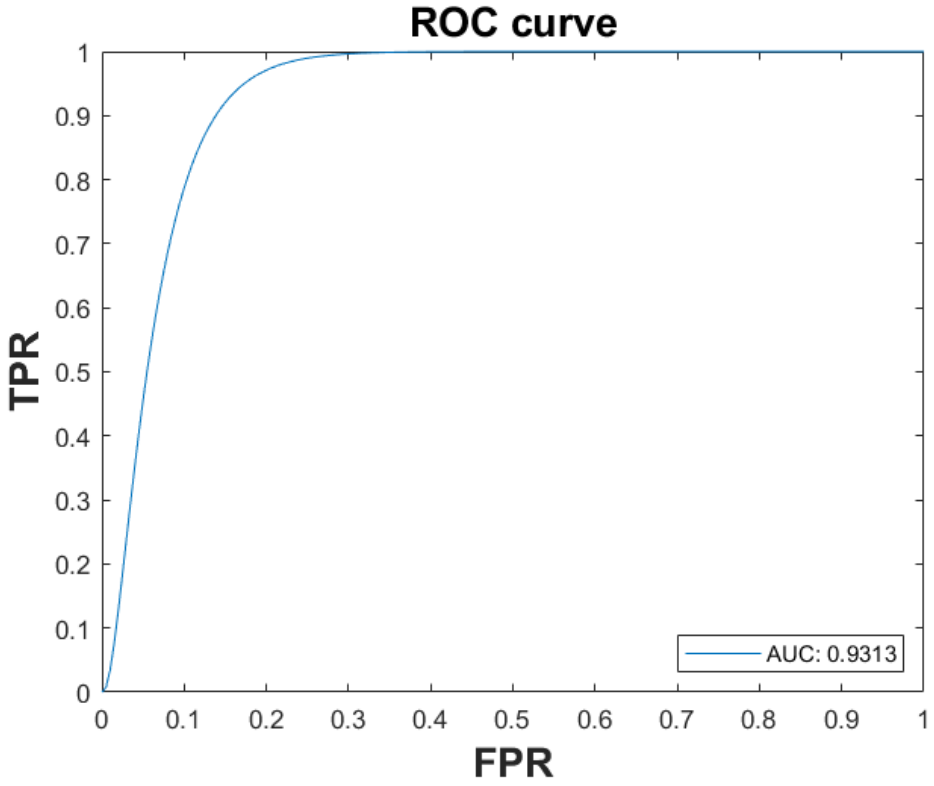


Figure 33. ROC curve for the DenseNet classifier.

5.2. Results

The test data was used for measuring the model performance, obtaining an AUC of 0.931 as shown in Figure 33. The ROC curve shows a very high level of performance, close to theoretically optimal, for the proposed classifier since the curve approaches near to the top-left corner of the graph. Table 8 shows the confusion matrix generated by the predictions of the DenseNet classifier.

The positive class refers to the irregular pattern presented to the left of Figure 30, labeled as "1" in the dataset, and the negative classes refers to the semi-concentric pattern labeled as "0", as indicated in Section 5.1.1. The values of the confusion matrix were used to calculate the accuracy of the model, as indicated in Section 2.2.2, obtaining an accuracy of 92%. Due to the novelty of using ^{18}F-FLT as a radiotracer on post-HSCT patients, to date there are no similar published works to compare this model performance against. Proper evaluation by medical staff will determine the feasibility of the proposed model for translation to clinical practice.

5.3. Discussion

In this chapter, I presented a CNN model for classifying the patterns generated by the spatial distribution of SUV within the bone marrow cavities of the vertebral bodies on post-HCST patients. The model uses the DenseNet architecture with four densely connected layers with batch normalization and ReLU activation. The patterns presented in Figure 30 served as criteria for generating the ground truth data, obtaining an accuracy of 92% after training. I expect that this model could be used as an auxiliary tool for monitoring the recovery process of HSCT patients.

Chapter 6. Conclusion

In this thesis, I have presented an automated framework for extracting SUV measurements from the undersampled PET/CT scans obtained on the study of post-HSCT patients [7],[8]. The proposed framework combines traditional image processing with the widely used convolutional neural networks for segmentation of bone structures and identifying individual vertebral bodies, which are used for obtaining the SUV from the patient scans, simplifying the time-consuming task of manually examining numerous scans. Additionally, a classifier was trained based on the spatial distribution of the SUV, which can be employed as a monitoring tool to assess patient status after transplant.

To accomplish the segmentation of bone structures from the CT scans present in the HSCT dataset, I trained a 3D U-Net variant of the architecture presented in [4]. The refinements that I introduced in my application include substituting the instance normalization by batch normalization in both the encoder and decoder path of the network, since the batch size I used during training was small due to memory constraints, and the addition of a stride convolution on the residual units for matching the input sizes when required, as described in the documentation provided by the MONAI framework [119].

The data used for training the network was provided by Nguyen [13], consisting of dictionary-based files for MATLAB® and raw slices containing the ground-truth annotations for three bone structures: vertebral body, sternum, and pelvis, where most of the annotations correspond to background voxels. I converted the annotations to the NIfTI format for better organization, and applied augmentations during the model training to increment the data

variability. To assess the imbalance of background and non-background voxels, I opted for the Dice Loss [80] as the loss function. The Dice scores obtained after training were slightly above 0.8 using a threshold of 0.5 on the generated probability map. Changing the threshold function to argmax resulted in an improvement of over +0.05 points in the Dice score, obtaining a mean value of 0.916 for the segmentation of vertebral bodies. The qualitative results presented in Section 3.2.2 indicate that the major discrepancies between the ground-truth and predicted volumes were caused by the apparition of false positive voxels, which are more visible in the intervertebral discs from the lumbar region for the vertebral body class, and in the costal cartilage for the sternum.

In order to identify the individual vertebrae from the segmented mask obtained from the 3D U-Net, I introduced an instance segmentation algorithm taking advantage of some anatomical priors. Padding the spinal column mask in the axial plane was performed to identify the presence of pedicles on each axial slice, which were used as indicator of the starting point of a new vertebra. The inter-vertebral boundaries were estimated introducing a parametric line between two contiguous vertebrae. I tested the algorithm using downsampled volumes from the Verse dataset [112]-[114], obtaining a mean Dice score of 0.898, 0.838 and 0.882 for the cervical, thoracic, and lumbar vertebrae respectively. I then ran the algorithm on the CT scans from the HSCT study [7], [8], corresponding to the 28^{th} day after the transplant, to segmentate the vertebrae from C2 to L5.

Segmentation of individual vertebrae was necessary to extract the SUV measurements from the bone marrow cavities. Since the PET scans and the CT scans from the HCST study present different resolutions, spacing and origin, I translated the segmented volumes from the CT space to the PET

space, using affine matrixes containing the spatial information. The PET volumes remained unaltered to prevent distorting the precalculated SUV. I then used the extracted values to calculate the median, mean, maximum and standard deviation of SUV within each individual vertebra. The results obtained with this method were compared to the method proposed by Carson [15] for instance segmentation of the vertebral bodies, resulting in a strong agreement in the values of standard deviation and maximum SUV, with most of the data concentrated within the ±2 margin. The variability in the mean and median SUV can be attributed to the differences in the methodology used for detecting the boundaries between contiguous vertebrae.

Finally, the 3D classifier presented in Chapter 5 was trained for classifying the patterns generated by the spatial distribution of the SUV within the bone marrow of the vertebral bodies. It has been suggested that the irregular pattern presented in Figure 30 could be potentially used as indicator of relapse on post-HSCT patients [12]. The segmented vertebral bodies obtained by using the method described in Chapter 4 were labelled according to the patterns discussed in Chapter 5. The selected model was based on the DenseNet architecture [16], with four densely connected blocks. After training, an AUC of 0.931 was obtained by the classifier, with an accuracy of 92%. Proper evaluation by medical staff will determine the feasibility of the proposed model for translation to clinical practice.

6.1. Original Contributions

The original contributions of this work include the following:

- A CNN for segmentation of bone structures on undersampled CT scans: the model is based on a 3D U-Net architecture and was trained using the HSCT dataset for segmentation of the sternum, pelvis, and vertebral bodies. Compared to other methods, this one was trained specifically for

scans with a low axial resolution and requires less training than the multi-view ensemble U-Net presented by Carson [15], obtaining a mean Dice score of 0.916 for segmentation of vertebral bodies.

- An algorithm for segmentation of the individual vertebral bodies: by using prior knowledge of the anatomical characteristics of the vertebrae, I introduced an iterative algorithm for identifying each vertebral body on the segmentation mask obtained from the 3D U-Net. The obtained mean Dice score for undersampled volumes is > 0.83 on the cervical, thoracic, and lumbar regions. The individual vertebrae were used for extracting the SUV from the PET scans, with results for the max SUV within ± 2 compared to Carson's method for SUV extraction [15].

- A 3D classifier for post-HSCT patients: by using the observations made on the spatial distribution of SUV values 28 days after the HSCT treatment, I trained a DenseNet [16] model for detecting possible relapsing cases, obtaining an AUC value of 0.931 and an accuracy of 92%. This model could assist on monitoring the recovery process of HSCT patients.

These contributions were specially tailored for the undersampled PET/CT scans obtained in the study of ^{18}F-FLT as radiotracer on post-HCST patients [7], [8] and I expect could facilitate the tasks performed by the physicians.

6.2. Recommendations for Further Research

Based on the work presented in this thesis, enhancements and future works include:

- Increasing the number of annotated volumes: although the 3D U-Net presents a high Dice Score (0.916 for vertebral bodies) using the available sparse annotations, the current model performance can be

improved by providing more training data using transfer learning. Additional data also contributes to a proper validation/test ratio.

- Increasing the number of segmentation classes (like femur, liver, and spleen) can contribute to future studies on localized regions. SUV extraction from the additional classes could potentially provide a better understanding of the patient behavior during the post-HSCT recovery process.

- The low axial resolution issue could be addressed with aid of Generative Adversarial Networks (GAN) for attempting to reconstruct the slices with a different Kernel, like the approach presented in [140]. This way, the image contrast can be enhanced without exposing the patients to higher radiation dose.

- The approach used in Chapter 4 provides a somewhat simplified method for estimating the vertebral boundaries given the limited axial resolution, which assumes no malformations on the patient's spine. Designing a proper geometrical model is required for evaluating patients presenting fractures, osteoporosis, or other pre-existing conditions.

- Additional CNN architectures can be trained as classifiers of the spatial distribution of SUV and to provide a performance comparison with the architecture presented in Chapter 5. Proper evaluation from medical specialists is still required for providing the ground truth data and determining the feasibility of the models.

Appendix A. Multi-class Segmentation Masks

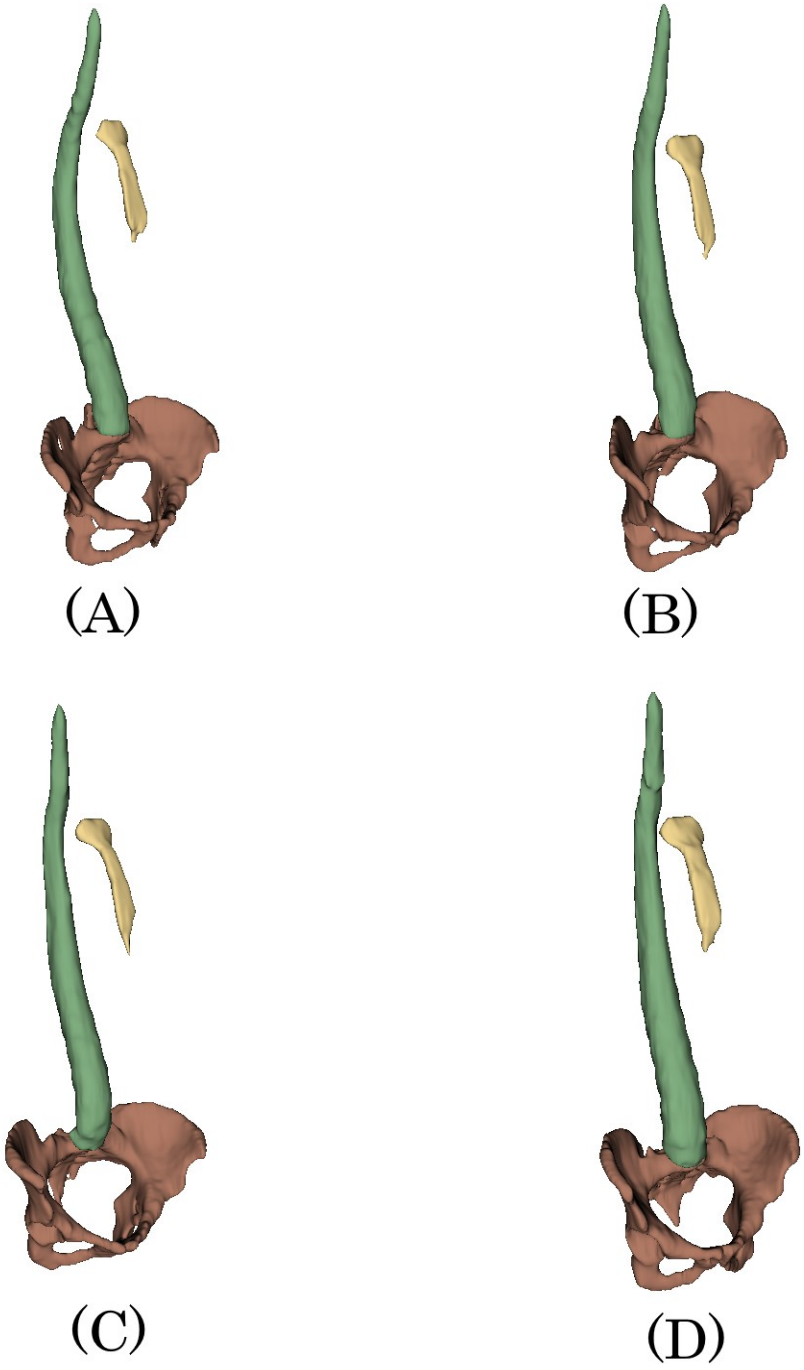


Figure 34. Examples of the segmentation masks obtained by using the argmax function and the U-Net described in Chapter 3, for three predicted classes: vertebral body (green), sternum (yellow), and pelvis (red) on four sample patients form the HSCT dataset.

Appendix B. Instance Segmentation of Vertebral Bodies

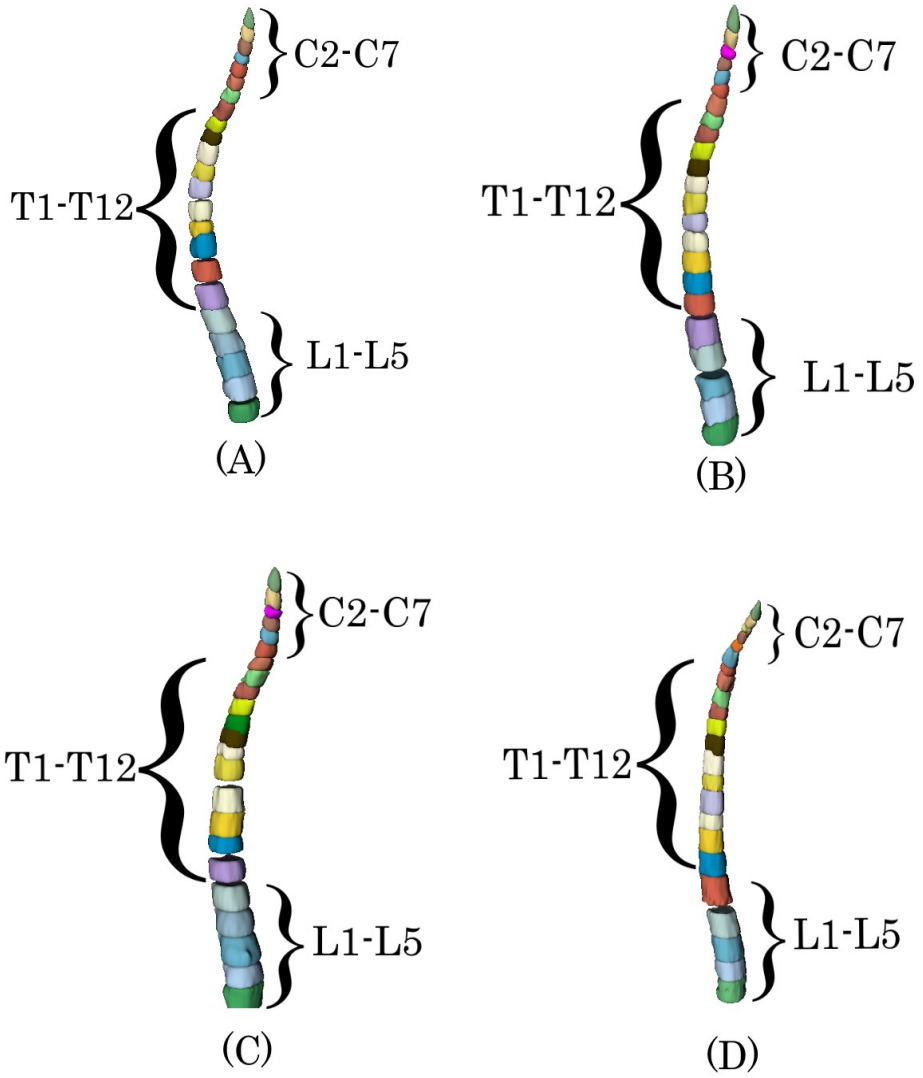
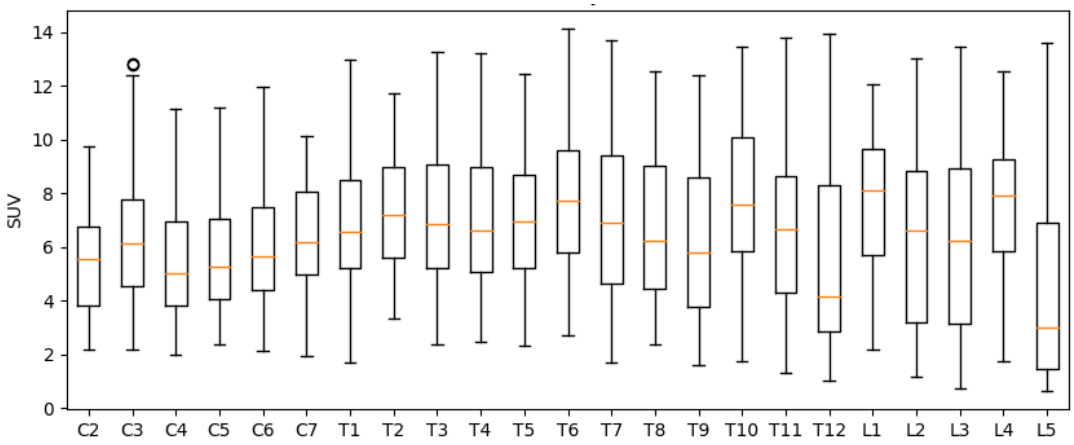
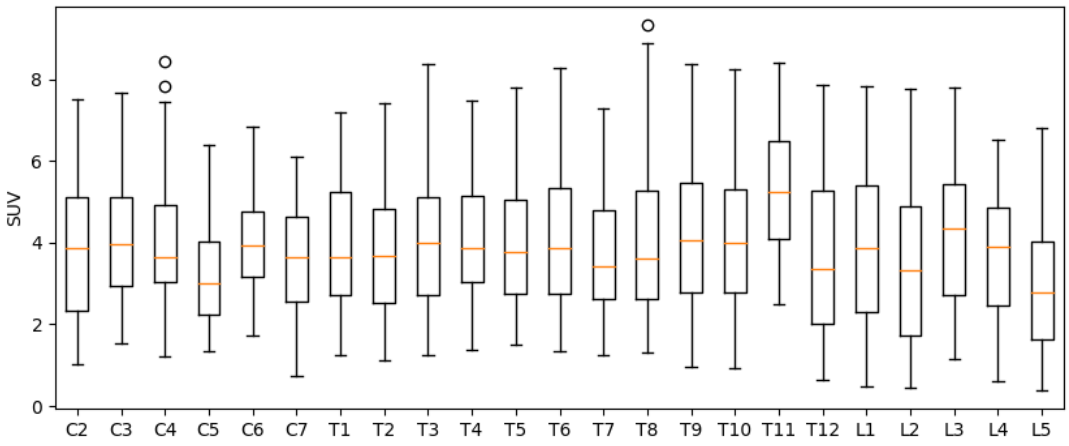


Figure 35. 3D rendering of the vertebral segmentation, obtained by using the method described in Chapter 4 on four sample patients from the HSCT dataset.

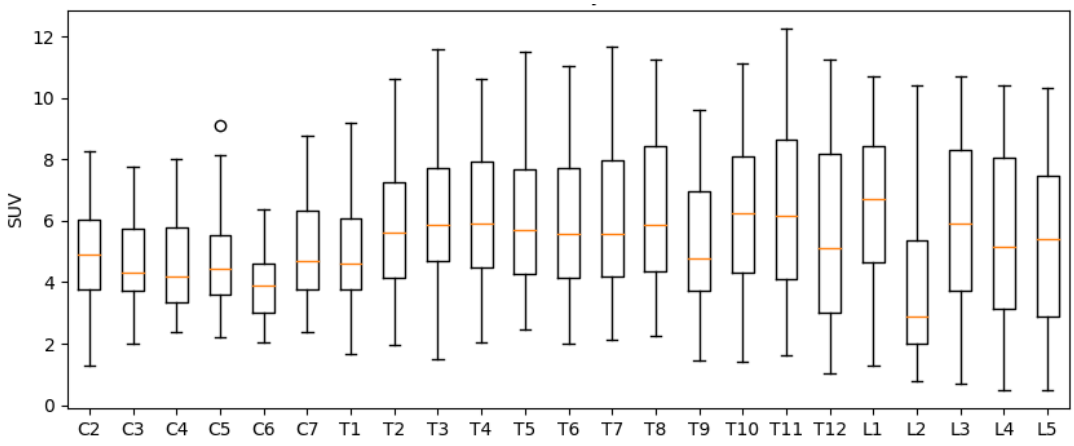
Appendix C. SUV Distribution Plots



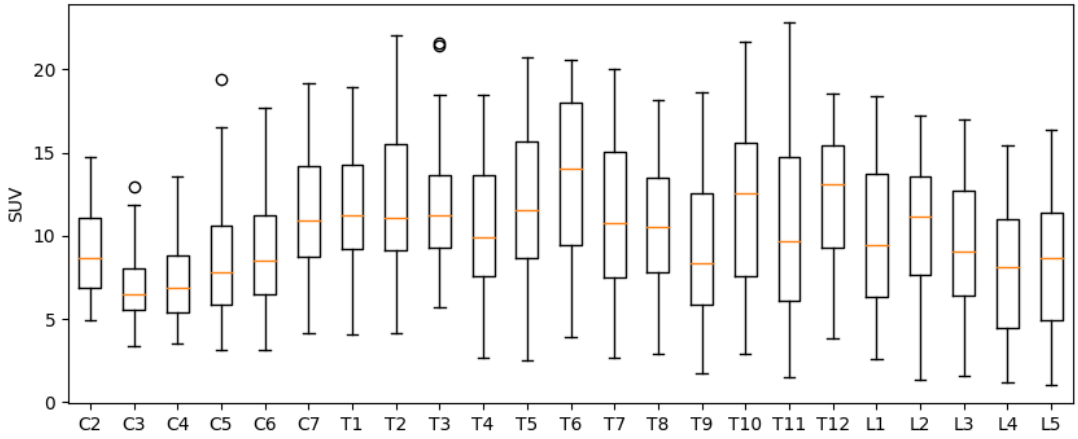
(A)



(B)



(C)



(D)

Figure 36. SUV distribution of the vertebral bodies from four sample patients (A to D) from the HSCT dataset on the 28th day after HSCT treatment, using the method described in Section 4.4. The boxes extend from the first quartile to the third quartile. Orange segments inside each box indicate the median value. Whiskers extend to the minimum/maximum value within ± 1.5 times the interquartile range. The circles represent values outside that range.

References

[1] FDA (2018). Medical Imaging. Retrieved Nov. 2021 from https://www.fda.gov/radiation-emitting-products/radiation-emitting-products-and-procedures/medical-imaging.

[2] Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., & Pal, C. (2016) The Importance of Skip Connections in Biomedical Image Segmentation. In *Carneiro G. et al. (eds) Deep Learning and Data Labeling for Medical Applications. DLMIA 2016, LABELS 2016. Lecture Notes in Computer Science*, vol 10008. Springer, Cham.

[3] Jadon, S. (2020, October). A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1-7. IEEE.

[4] Kerfoot, E., Clough, J., Oksuz, I., Lee, J., King, A. P., & Schnabel, J. A. (2019) Left-Ventricle Quantification Using Residual U-Net. In *Pop M. et al. (eds) Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges. STACOM 2018. Lecture Notes in Computer Science*, vol 11395. Springer, Cham.

[5] Ronneberger, O., Fischer, P. & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241. Springer, Cham.

[6] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In *International conference on medical image computing and computer-assisted intervention*, pp. 424-432. Springer, Cham.

[7] Williams, K. M., Holter-Chakrabarty, J., Lindenberg, L., Duong, Q., Vesely, S. K., Nguyen, C. T., Havlicek, J. P., Kurdziel, K., Gea-Banacloche, J., Lin, F. I., Avila, D. N., Selby, G., Kanakry, C. G., Li, S., Scordino, T., Adler, S., Bollard C. M., Choyke, P., & Gress, R. E. (2018) Imaging of subclinical haemopoiesis after stem-cell transplantation in patients with haematological malignancies: a prospective pilot study. In *Lancet Haematology*, 5(1), pp. 44-52.

[8] Williams, K. M., & Chakrabarty, J. H. (2020). Imaging haemopoietic stem cells and microenvironment dynamics through transplantation. In *The Lancet. Haematology*, 7(3), pp. 259-269.

[9] Mattsson, J., Ringdén, O., & Storb, R. (2008). Graft failure after allogeneic hematopoietic cell transplantation. In *Biology of blood and marrow*

transplantation: journal of the American Society for Blood and Marrow Transplantation, 14(1 Suppl 1), pp. 165-170.

[10] Zahid, M. (2015). Methods of reducing pain during bone marrow biopsy: a narrative review. In *Annals of Palliative Medicine*, 4(4), pp. 184-193.

[11] Luiz Vieira, J. S., da Silva Herrero, C. F., Porto, M. A., Nogueira Barbosa, M. H., Garcia, S. B., Zambelli Ramalho, L. N., & Aparecido Defino, H. L. (2015). Evaluation of terminal vertebral plate on cervical spine at different age groups and its correlation with intervertebral disc thickness. In *Revista brasileira de ortopedia*, 44(1), pp. 20-25.

[12] Holter, J. (2021). Personal communication.

[13] Nguyen, C. (2020) Personal communication.

[14] NiBabel. (2020, November 28). Coordinate systems and affines. Retrieved Sept. 2021 from https://nipy.org/nibabel/index.html.

[15] Carson, B. (2021). Automatic Bone Structure Segmentation of Under-sampled CT/FLT-PET Volumes for HSCT Patients. [M.S. Thesis, University of Oklahoma]. SHAREOK. Retrieved Sept. 2021 from https://hdl.handle.net/11244/330194.

[16] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708.

[17] Buzug, T. M. (2010). Computed tomography: From photon statistics to modern cone-beam CT. Berlin: Springer.

[18] Seeram, E. (2015). Computed Tomography - E-Book: Physical Principles, Clinical Applications, and Quality Control. United States: Elsevier Health Sciences.

[19] Bushberg, J. T., Siebert, J. A., Leidholdt, E. M., & Boone, J. M. (2012). The essential physics of medical imaging. Philadelphia: Wolters Kluwer.

[20] Dalrymple, N., Prasad, S., El-Merhi, F., & Chintapalli, K. (2007). Price of Isotropy in Multidetector CT. In *RadioGraphics*, 27(1), pp. 49-62.

[21] Dalrymple, N., Prasad, S., Freckleton, M., & Chintapalli, K. (2005). Introduction to the Language of Three-dimensional Imaging with Multidetector CT. In *RadioGraphics*, 25(5), pp. 1409-1428.

[22] Wernick, M. N., & Aarsvold, J. N. (2004). Emission Tomography: The Fundamentals of PET and SPECT. Netherlands: Elsevier Science.

[23] Granov, A., Tiutin, L., & Schwarz, T. (2013). Positron Emission Tomography. Springer-Verlag Berlin Heidelberg.

[24] Bailey, D. L., Townsend, D. W, Valk, P. E, & Maisey, M. N. (2005). Positron Emission Tomography: Basic Science and Clinical Practice. Springer.

[25] Tahari, A., Chien, D., Azadi, J., & Wahl, R. (2014). Optimum Lean Body Formulation for Correction of Standardized Uptake Value in PET Imaging. In *Journal of Nuclear Medicine*, 55(9), pp. 1481-1484.

[26] Highsmith, Jason M. (n.d) Spinal Anatomy. Retrieved Sept. 2021 from https://www.spineuniverse.com/anatomy.

[27] Kuri, J., & Stapleton, E. (2002). The Spine at Trial: Practical Medicolegal Concepts about the Spine. United States: American Bar Association.

[28] Garfin, S., Eismont, F., Bell, G., Bono, C., & Fischgrund, J. (2017). Rothman-Simeone The Spine. United States: Elsevier Health Sciences.

[29] Birbrair, A., & Frenette, P. S. (2016). Niche heterogeneity in the bone marrow. In *Annals of the New York Academy of Sciences*, 1370(1), pp. 82-96.

[30] Miller-Keane. (2003). Encyclopedia and Dictionary of Medicine, Nursing, and Allied Health, Seventh Edition. Retrieved Sept. 2021 from https://medical-dictionary.thefreedictionary.com.

[31] Park, B., Yoo, K. H., & Kim, C. (2015). Hematopoietic stem cell expansion and generation: the ways to make a breakthrough. In *Blood Research*, 50(4), pp. 194-203.

[32] Leung, K. (2005). [18F]Fluoro-2-deoxy-2-D-glucose. In: *Molecular Imaging and Contrast Agent Database (MICAD) [Internet]. Bethesda (MD): National Center for Biotechnology Information (US);* 2004-2013. Retrieved Sept. 2021 from: https://www.ncbi.nlm.nih.gov/books/NBK23335/.

[33] Leung, K. (2005) 3'-Deoxy-3'-[18F]fluorothymidine. In: *Molecular Imaging and Contrast Agent Database (MICAD) [Internet]. Bethesda (MD): National Center for Biotechnology Information (US);* 2004-2013. Retrieved Sept. 2021 from: https://www.ncbi.nlm.nih.gov/books/NBK23373/.

[34] Long, B. W., Rollins, J. H., Smith, B. J. (2018). Merrill's Atlas of Radiographic Positioning and Procedures Volume 1. United States: Elsevier Health Sciences.

[35] Slicer Wiki. (2014). Coordinate systems. Retrieved Sept. 2021 from https://www.slicer.org/w/index.php?title=Coordinate_systems.

[36] Abbena, E., Salamon, S., & Gray, A. (2017). Modern Differential Geometry of Curves and Surfaces with Mathematica. United States: CRC Press.

[37] Woods, F. S. (2013). Higher Geometry: An Introduction to Advanced Methods in Analytic Geometry. Dover Publications.

[38] Bidgood, W. D., Jr, Horii, S. C., Prior, F. W., & Van Syckle, D. E. (1997). Understanding and using DICOM, the data interchange standard for biomedical imaging. In *Journal of the American Medical Informatics Association: JAMIA*, 4(3), 199-212.

[39] The Medical Imaging Technology Association. (2021). DICOM Standard. Retrieved Sept. 2021 from https://www.dicomstandard.org/current.

[40] Neuroimaging Informatics Technology Initiative (2013, December 18). NIfTI. Retrieved Sept. 2021 from https://nifti.nimh.nih.gov/.

[41] Whitcher, B., Schmid, V. J., & Thorton, A. (2011). Working with the DICOM and NIfTI Data Standards in R. In *Journal of Statistical Software*, 44(6), pp. 1-29.

[42] Jiang, X., & Suri, J. S. (2016). Biomedical Image Segmentation: Advances and Trends. United States: CRC Press.

[43] Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2018). Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9404-9413.

[44] Zaitoun, N. M, & Aqel M. J. (2015). Survey on Image Segmentation Techniques. In *Procedia Computer Science*, 65, pp. 797-806.

[45] Gonzalez, R. C., & Woods, R. E. (2008). Digital image processing. Second edition. Italy: Pearson/Prentice Hall.

[46] Mostafavi, H., Sloutsky, A., & Jeung, A. (2012). Detection and localization of radiotherapy targets by template matching. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6023-6027.

[47] Huangfu, M., Konaka, S., Akutagawa, M., & Emoto, T. (2012). The improved matching method to cell extract using ellipse template. In *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pp. 325-328.

[48] Acton, S. T., Ray, N. (2009). Biomedical Image Analysis: Segmentation. United States: Morgan & Claypool Publishers.

[49] Zhou, S. K. (2015) Medical Image Recognition, Segmentation and Parsing: Machine Learning and Multiple Object Approaches. Netherlands: Elsevier Science.

[50] Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. In *BMC Medical Imaging 15*.

[51] Crum, W. R., Camara, O., & Hill, D. L. G. (2006). Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis. In *IEEE Transactions on Medical Imaging*, vol. 25, no. 11, pp. 1451-1461.

[52] Krig, S. (2014). Computer Vision Metrics: Survey, Taxonomy, and Analysis. United States: Apress.

[53] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. In *Journal of Machine Learning Technologies*, vol. 2, pp. 37-63.

[54] Fenster, A., & Chiu, B. (2005) Evaluation of Segmentation algorithms for Medical Imaging. *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 7186-7189.

[55] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[56] Theodoridis, S. (2020). Machine Learning: A Bayesian and Optimization Perspective. United Kingdom: Elsevier Science.

[57] Bishop, C. M. (2013). Pattern Recognition and Machine Learning: All "just the Facts 101" Material. India: Springer (India) Private Limited.

[58] Chapelle, O., Scholkopf, B., & Zien, A. (2010). Semi-supervised learning. Cambridge, Mass: MIT.

[59] Baldi, P., & Brunak, S. (2001). Bioinformatics: The machine learning approach. Cambridge, Mass: MIT Press.

[60] Ong, C. S., Deisenroth, M. P., & Faisal, A. (2020). Mathematics for Machine Learning. United Kingdom: Cambridge University Press.

[61] Kohli, M. D., Summers, R. M., & Geis, J. R. (2017). Medical Image Data and Datasets in the Era of Machine Learning-Whitepaper from the 2016 C-MIMI Meeting Dataset Session. In *Journal of digital imaging*, 30(4), pp. 392-399.

[62] McCulloch, W.S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. In *Bulletin of Mathematical Biophysics 5*, pp. 115-133.

[63] Trappenberg, T. (2019). Fundamentals of Machine Learning. United Kingdom: OUP Oxford.

[64] Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning, neural and statistical classification. New York: Ellis Horwood.

[65] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[66] Maalej, R., & Kherallah, M. (2020). Improving the DBLSTM for on-line Arabic handwriting recognition. In *Multimedia Tools and Applications* 79, pp. 17969-17990.

[67] Nielsen, M. A. (2015). Neural Networks and Deep Learning. Determination Press.

[68] Lydia, A. & Francis, Sagayaraj. (2019). A Survey of Optimization Techniques for Deep Learning Networks. In *International Journal for Research in Engineering Application & Management (IJREAM)*, vol. 5, Issue 02.

[69] Kastrati, M. & Biba, Marenglen. (2021). A state-of-the-art survey of advanced optimization methods in machine learning. In *International Journal of Computer Science and Information Security (IJCSIS)*, 19(7).

[70] Diederik P. Kingma, & Jimmy Ba. (2017). Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR).*

[71] Ioffe, S. & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International conference on machine learning*, pp. 448-456. PMLR.

[72] Awais, M., Iqbal, M. T. B. & Bae S. H. (2020). Revisiting Internal Covariate Shift for Batch Normalization. In *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 5082-5092.

[73] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* pp. 8024-8035. Curran Associates, Inc. Retrieved Sept. 2021 from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[74] TensorFlow Developers. (2021). TensorFlow (v2.3.3). Retrieved Sept. 2021 from https://github.com/tensorflow/tensorflow.

[75] Fan, C. (2016). Survey of convolutional neural network. Retrieved Sept. 2021 from https://fanchenyou.github.io/homepage/docs/cnn_survey.pdf.

[76] Badrinarayanan, V., Handa, A. & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling. In *IEEE transactions on pattern analysis and machine intelligence*, 39(12), pp. 2481-2495.

[77] Yasrab, R., Gu, N. & Zhang, X. (2017). An encoder-decoder based Convolution Neural Network (CNN) for future Advanced Driver Assistance System (ADAS). In *Applied Sciences*, Volume 7(4).

[78] Ciresan, D., Giusti, A., Gambardella, L., & Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[79] Arganda-Carreras, I., Turaga, S., Berger, D., Cireşan, D., Giusti, A., Gambardella, L., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J., Liu, T., Seyedhosseini, M., Tasdizen, T., Kamentsky, L., Burget, R., Uher, V., Tan, X., Sun, C., Pham, T., Bas, E., Uzunbas, M., Cardona, A., Schindelin, J., & Seung, H. (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. In *Frontiers in Neuroanatomy*, 9, 142.

[80] Milletari, F., Navab, N. & Ahmadi, S. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pp. 565-571. IEEE.

[81] Zhang, J., Du, J., Liu, H., Hou, X., Zhao, Y. & Ding, M. (2019). LU-NET: An Improved U-Net for Ventricular Segmentation. *IEEE Access,* vol. 7, pp. 92539-92546.

[82] Tran, P. V. (2016) A fully convolutional neural network for cardiac segmentation in short-axis MRI. arXiv:1604.00494.

[83] Xu, Z.; Wu, Z.; Feng, J. (2018). CFUN: Combining faster R-CNN and U-net network for efficient whole heart segmentation. arXiv:1812.04914.

[84] Chlebus, G., Schenk, A., Moltz, J. H., van Ginneken, B., Hahn, H. K., & Meine, H. (2018). Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing. In *Scientific reports*, 8(1), 15497.

[85] Christ, P. F., Elshaer, M. E. A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., ... & Menze, B. H. (2016). Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random

fields. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 415-423. Springer, Cham.

[86] Nie, D., Wang, L., Adeli, E., Lao, C., Lin, W., & Shen, D. (2019). 3-D Fully Convolutional Networks for Multimodal Isointense Infant Brain Image Segmentation. In *IEEE transactions on cybernetics,* 49(3), pp. 1123-1136.

[87] Tuan, T. A., Pham, T. B., Kim, J. Y., & Tavares, J. (2020). Alzheimer's diagnosis using deep learning in segmenting and classifying 3D brain MR images. In *The International journal of neuroscience*, pp. 1-10.

[88] Casamitjana A., Catà M., Sánchez I., Combalia M., & Vilaplana V. (2018) Cascaded V-Net Using ROI Masks for Brain Tumor Segmentation. In *Lecture Notes in Computer Science*, vol. 10670. Springer, Cham.

[89] Novikov, A. A., Lenis, D., Major, D., Hladůvka, J., Wimmer, M., & Bühler, K. (2018). Fully Convolutional Architectures for Multiclass Segmentation in Chest Radiographs. In *IEEE Transactions on Medical Imaging*, 37, pp. 1865-1876.

[90] Jue, J., Jason, H., Neelam, T., Andreas, R., Sean, B. L., Joseph, D. O., & Harini, V. (2019). Integrating cross-modality hallucinated MRI with CT to aid mediastinal lung tumor segmentation. In *Medical image computing and computer-assisted intervention: MICCAI. International Conference on Medical Image Computing and Computer-Assisted Intervention*, 11769, pp. 221-229.

[91] Anthimopoulos, M., Christodoulidis, S., Ebner, L., Geiser, T., Christe, A., & Mougiakakou, S. G. (2019). Semantic Segmentation of Pathological Lung Tissue with Dilated Fully Convolutional Networks. In *IEEE Journal of Biomedical and Health Informatics*, 23, pp. 714-722.

[92] Simonyan, K. & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

[93] Xi, P., Shu, C., & Goubran, R. (2018). Abnormality Detection in Mammography using Deep Convolutional Neural Networks. In *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1-6.

[94] Wang, W., Chakraborty, G., & Chakraborty, B. (2020). 3D Multi-scale DenseNet for Malignancy Grade Classification of Pulmonary Nodules. In *2020 11th International Conference on Awareness Science and Technology (iCAST)*, pp. 1-4.

[95] Demir, A., & Yilmaz, F. (2020). Inception-ResNet-v2 with Leakyrelu and Average pooling for More Reliable and Accurate Classification of Chest X-ray Images. In *2020 Medical Technologies Congress (TIPTEKNO)*, pp. 1-4.

[96] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng. A. Y. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. arXiv:1711.05225.

[97] Saito, K., Zhao, Y., & Zhong, J. (2019). Heart Diseases Image Classification Based on Convolutional Neural Network. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 930-935.

[98] Zheng, Z., Zhang, H., Li, X., Liu, S., & Teng, Y. (2021). ResNet-Based Model for Cancer Detection. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 325-328.

[99] Wang, M., & Gong, X. (2020). Metastatic Cancer Image Binary Classification Based on Resnet Model. In *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pp. 1356-1359.

[100] Berstad, T., Riegler, M., Espeland, H., Lange, T., Smedsrud, P., Pogorelov, K., Kvale Stensland, H., & Halvorsen, P. (2018). Tradeoffs Using Binary and Multiclass Neural Network Classification for Medical Multidisease Detection. In *2018 IEEE International Symposium on Multimedia (ISM)*, pp. 1-8.

[101] Yuan, L. (2020). A brief history of deep learning frameworks. Retrieved Oct. 2021 from https://syncedreview.com/2020/12/14/a-brief-history-of-deep-learning-frameworks/.

[102] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675-678.

[103] Facebook Open Source. (2018). Caffe2 and PyTorch join forces to create a Research + Production platform PyTorch 1.0. Retrieved Oct. 2021 from https://caffe2.ai/blog/2018/05/02/Caffe2_PyTorch_1_0.html.

[104] The MathWorks, Inc. (2021). Deep Learning Toolbox. Retrieved Oct. 2021 from https://www.mathworks.com/help/deeplearning/index.html.

[105] PyTorch (2021). Ecosystem Tools. Retrieved Oct. 2021 from https://pytorch.org/ecosystem/.

[106] Yao, J., Burns, J. E., Forsberg, D., Seitel, A., Rasoulian, A., Abolmaesumi, P., Hammernik, K., Urschler, M., Ibragimov, B., Korez, R., Vrtovec, T., Castro-Mateos, I., Pozo, J. M., Frangi, A. F., Summers, R. M., & Li, S. (2016). A multi-center milestone study of clinical vertebral CT segmentation. In *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society*, 49, pp. 16-28.

[107] Forsberg, D. (2014). Atlas-Based Segmentation of the Thoracic and Lumbar Vertebrae. In *2nd MICCAI Workshops on Computational Methods and Clinical Applications for Spine Imaging (CSI2014)*. Boston, USA: Springer.

[108] Seitel A., Rasoulian A., Rohling R., & Abolmaesumi P. (2015) Lumbar and Thoracic Spine Segmentation Using a Statistical Multi-object Shape+Pose Model. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*. *Lecture Notes in Computational Vision and Biomechanics*, vol 20. Springer, Cham.

[109] Hammernik K., Ebner T., Stern D., Urschler M., & Pock T. (2015) Vertebrae Segmentation in 3D CT Images Based on a Variational Framework. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging. Lecture Notes in Computational Vision and Biomechanics*, vol 20. Springer, Cham.

[110] Korez R., Ibragimov B., Likar B., Pernuš F., & Vrtovec T. (2015) Interpolation-Based Shape-Constrained Deformable Model Approach for Segmentation of Vertebrae from CT Spine Images. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging. Lecture Notes in Computational Vision and Biomechanics*, vol 20. Springer, Cham.

[111] Castro-Mateos I., Pozo J.M., Lazary A., & Frangi A. (2015) 3D Vertebra Segmentation by Feature Selection Active Shape Model. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging. Lecture Notes in Computational Vision and Biomechanics, vol 20*. Springer, Cham.

[112] Sekuboyina, A., Husseini, M., Bayat, A., Löffler, M., Liebl, H., Li, H., Tetteh, G., Kukačka, J., Payer, C., Štern, D., & et al. (2021). VerSe: A Vertebrae labelling and segmentation benchmark for multi-detector CT images. In *Medical Image Analysis*, 73, 102166.

[113] Liebl, H., Schinz, D., Sekuboyina, A., Malagutti, L., Löffler, M. T., Bayat, A., Husseini, M. E., Tetteh, G., Grau, K., Niederreiter, E., Baum, T., Wiestler, B., Menze, B., Braren, R., Zimmer, C., & Kirschke, J. S. (2020). A Computed Tomography Vertebral Segmentation Dataset with Anatomical Variations and Multi-Vendor Scanner Data. arXiv: 2103.06360.

[114] Löffler, Maximilian & Sekuboyina, Anjany & Jacob, Alina & Grau, Anna-Lena & Scharr, Andreas & Husseini, Malek & Kallweit, Mareike & Zimmer, Claus & Baum, Thomas & Kirschke, Jan. (2020). A Vertebral Segmentation Dataset with Fracture Grading. In *Radiology: Artificial Intelligence*, 2.

[115] Altini, N., De Giosa, G., Fragasso, N., Coscia, C., Sibilano, E., Prencipe, B., & Bevilacqua, V. (2021). Segmentation and Identification of Vertebrae in CT scans Using CNN, k-Means Clustering and k-NN. In *Informatics*, 8(2), 40.

[116] Sekuboyina, A., Rempfler, M., Valentinitsch, A., Menze, B. H. & Kirschke, J. S. (2021). Labelling Vertebrae with 2D Reformations of Multidetector CT Images: An Adversarial Approach for Incorporating Prior Knowledge of Spine Anatomy. In *Radiology: Artificial Intelligence*, 2(2), e190074.

[117] Nguyen, C. T., Havlicek, J. P., Chakrabarty, J. H., Duong, Q. & Vesely, S. K. (2016). Towards automatic 3D bone marrow segmentation. In *2016 IEEE Southwest Symposium on Image Analysis and Interpretation, SSIAI 2016 – Proceedings*, pp. 9-12. Institute of Electrical and Electronics Engineers Inc.

[118] Nguyen, C. T., Havlicek, J. P., Duong, Q., Vesely, S., Gress, R., Lindenberg, L., Choyke, P., Chakrabarty, J., & Williams, K. M. (2016). An automatic 3D CT/PET segmentation framework for bone marrow proliferation assessment. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 4126-4130.

[119] MONAI Consortium. (2020). MONAI: Medical Open Network for AI (0.6.0). Retrieved Sept. 2021 from https://github.com/Project-MONAI/MONAI.

[120] Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J. C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., Buatti, J., Aylward, S., Miller, J. V., Pieper, S., & Kikinis, R. (2012). 3D Slicer as an image computing platform for the Quantitative Imaging Network. In *Magnetic resonance imaging*, 30(9), pp. 1323-1341.

[121] The MathWorks, Inc (2021). Volume Viewer. Retrieved Sept. 2021 from https://www.mathworks.com/help/images/ref/volumeviewer-app.html.

[122] Brett, M., Markiewicz, C. J., Hanke, M., Côté, M. A., Cipollini, B., McCarthy, P, … & Reddam, V. R. (2020). Nipy/nibabel: 3.2.1 (Version 3.2.1). Version 3.2.1. Retrieved Sept. 2021 from https://github.com/nipy/nibabel.

[123] Beare, R., Lowekamp, B., & Yaniv, Z. (2018). Image Segmentation, Registration and Characterization in R with SimpleITK. In *Journal of Statistical Software*, 86(8), pp. 1-35.

[124] Yaniv, Z., Lowekamp, B.C., Johnson, H.J. *et al.* (2018). SimpleITK Image-Analysis Notebooks: A Collaborative Environment for Education and Reproducible Research. In *Journal of Digital Imaging*, 31, pp. 290-303.

[125] Lowekamp, B., Chen, D., Ibanez, L., & Blezek, D. (2013). The Design of SimpleITK. In *Frontiers in Neuroinformatics*, 7, 45.

[126] Nilearn (2021). Machine learning for NeuroImaging in Python. Retrieved Sept. 2021 from. https://github.com/nilearn/nilearn.

[127] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. arXiv:1607.08022.

[128] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026-1034.

[129] Pytorch (2021). Optimizing Model Parameters. Retrieved Oct. 2021 from https://pytorch.org/tutorials/beginner/basics/optimization_tutorial.html.

[130] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pp. 437-478. Springer, Berlin, Heidelberg.

[131] Niculae, V., Martins, A., Blondel, M., & Cardie, C. (2018, July). Sparsemap: Differentiable sparse structured inference. In *International Conference on Machine Learning*, pp. 3799-3808. PMLR. arXiv: 1802.04223.

[132] Blumfield, A. & E. Blumfield (2014). Automated Vertebral Body Image Segmentation for Medical Screening, U.S patent US20130077840A12.

[133] Xia, L., Xiao, L., Quan, G., & Bo, W. (2020). 3D Cascaded Convolutional Networks for Multi-vertebrae Segmentation. In *Current medical imaging*, 16(3), pp. 231-240.

[134] Spinal Cord Injury Information Pages. (2021) Anatomy of the spine. Retrieved Oct. 2021 from https://www.sci-info-pages.com/anatomy-of-the-spine/.

[135] SpineWeb iniciative. (2018). Datasets. Retrieved Nov. 2021 from http://spineweb.digitalimaginggroup.ca/Index.php?n=Main.Datasets.

[136] Hamberg, L. M., Hunter, G. J., Alpert, N. M., Choi, N. C., Babich, J. W., & Fischman, A. J. (1994). The dose uptake ratio as an index of glucose metabolism: useful parameter or oversimplification?. In *Journal of nuclear medicine: official publication, Society of Nuclear Medicine*, 35(8), pp. 1308-1312.

[137] Thie, J. A. (2004). Understanding the standardized uptake value, its methods, and implications for usage. In *Journal of nuclear medicine: official publication, Society of Nuclear Medicine*, 45(9), 1431-1434.

[138] Lodge, M. A., Holdhoff, M., Leal, J. P., Bag, A. K., Nabors, L. B., Mintz, A., Lesser, G. J., Mankoff, D. A., Desai, A. S., Mountz, J. M., Lieberman, F. S., Fisher, J. D., Desideri, S., Ye, X., Grossman, S. A., Schiff, D., & Wahl, R. L. (2017). Repeatability of ^{18}F-FLT PET in a Multicenter Study of Patients with High-Grade Glioma. In *Journal of nuclear medicine: official publication, Society of Nuclear Medicine*, 58(3), pp. 393-398.

[139] Mogensen, M. B., Loft, A., Aznar, M., Axelsen, T., Vainer, B., Osterlind, K., & Kjaer, A. (2017). FLT-PET for early response evaluation of colorectal cancer patients with liver metastases: a prospective study. In *EJNMMI research*, 7(1), 56.

[140] Yang, S., Kim, E. Y., & Ye, J. C. (2021). Continuous Conversion of CT Kernel using Switchable CycleGAN with AdaIN. In *IEEE Transactions on Medical Imaging (2021)*.