UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

AUTOMATIC BONE STRUCTURE SEGMENTATION OF UNDER-SAMPLED CT/FLT-PET VOLUMES FOR HSCT PATIENTS

A THESIS
SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

Brandon D. Carson
Norman, Oklahoma
2021

AUTOMATIC BONE STRUCTURE SEGMENTATION OF UNDER-SAMPLED CT/FLT-PET VOLUMES FOR HSCT PATIENTS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Joseph Havlicek, Chair

Dr. Samuel Cheng

Dr. Bin Zheng

© Copyright by Brandon D. Carson 2021
All Rights Reserved.

Acknowledgements

I must thank Dr. Joseph Havlicek, my thesis advisor, for developing my interest in image processing, for his always-candid guidance, and for offering me the opportunity to work on such an interesting and worthwhile research project.

All the committee members have my sincere gratitude for their time and consideration in reviewing this thesis.

I want to recognize Dr. Chuong Nguyen, for providing the labelled training data used in this thesis and for the discussions in the early stages of this project.

I am grateful to 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 who allowed my research on their novel HSCT patient image data. Dr. Holter-Chakrabarty was also supreme help in answering questions about her ongoing HSCT research.

Thanks to the National Cancer Institute, NIH, for funding the study that included acquisition of the HSCT patient data used in this thesis under contract number HHSN261200800001E.

Finally, I attribute the existence of this thesis to the wellspring of support and encouragement I have received from my family and friends.

Table of Contents

Abstract

In this thesis I present a pipeline for the instance segmentation of vertebral bodies from joint CT/FLT-PET image volumes that have been purposefully under-sampled along the axial direction to limit radiation exposure to vulnerable HSCT patients. The under-sampled image data makes the segmentation of individual vertebral bodies a challenging task, as the boundaries between the vertebrae in the thoracic and cervical spine regions are not well resolved in the CT modality, escaping detection by both humans and algorithms. I train a multi-view, multi-class U-Net to perform semantic segmentation of the vertebral body, sternum, and pelvis object classes. These bone structures contain marrow cavities that, when viewed in the FLT-PET modality, allow us to investigate hematopoietic cellular proliferation in HSCT patients non-invasively. The proposed convnet model achieves a Dice score of 0.9245 for the vertebral body object class and shows qualitatively similar performance on the pelvis and sternum object classes. The final instance segmentation is realized by combining the initial vertebral body semantic segmentation with the associated FLT-PET image data, where the vertebral boundaries become well-resolved by the 28th day post-transplant. The vertebral boundary detection algorithm is a hand-crafted spatial filter that enforces vertebra span as an anatomical prior, and it performs similar to a human for the detection of all but one vertebral boundary in the entirety of the HSCT patient dataset. In addition to the segmentation model, I propose, design, and test a "drop-in" replacement up-sampling module that allows state-of-the-art super-resolution convnets to be used for purely asymmetric upscaling tasks (tasks where only one image dimension is scaled while the other is held to unity). While the asymmetric SR convnet I develop falls short of the initial goal, where it was to be used to enhance the unresolved vertebral boundaries of the under-sampled CT image data, it does objectively upscale medical image data more accurately than naïve interpolation methods and may be useful as a pre-processing step for other medical imaging tasks involving anisotropic pixels or voxels.

Chapter 1
Introduction

Over the past decade the convolutional neural network has proliferated to become an essential component – and often *the* essential component – in state-of-the-art solutions to certain digital image processing and computer vision tasks such as object detection [1] and image segmentation [2]. In this thesis I deploy a few of these networks towards a specific application in the medical imaging domain: the segmentation of individual vertebral bodies, pelvis, and sternum bone structures from under-sampled dual-modality CT/PET volumes of hematopoietic stem cell transplant (HSCT) patients. In one context, "under-sampled" means these image volumes have a lower spatial sampling rate in one dimension than the others, creating non-cubic voxels. In another context, "under-sampled" means that the image volumes are not sampled at a high enough rate to adequately represent the phenomena of interest. Both contexts apply to this problem. CT and PET medical image volumes are captured by taking a sequence of 2D scans, typically along the length of the body. Commonly, the sampling rate along the length of the body matches the span of the pixels in the 2D scans, creating isotropic voxels that represent discrete cubic regions of space [3]. Sometimes, however, it is necessary to limit the radiation exposure to a vulnerable patient, such as one undergoing hematopoietic stem cell transplantation [4]. One way to limit radiation exposure is by under-sampling along the length of the body, which results in anisotropic voxels in the shape of rectangular prisms instead of cubes. Under-sampling in this way presents a set of unique challenges to the medical imaging practitioner undertaking a task that, by typical methods, requires a higher resolution. In many respects overcoming the obstacle of low resolution is the premise of this thesis: I show in Chapter 3 that even a human has trouble detecting the individual boundaries between the vertebrae using under-sampled CT image volume data; in Chapter 4 I attempt to increase the resolution along the under-sampled axis by developing a method for asymmetric super-resolution; and in Chapter 5, I overcome the low resolution by using PET image volumes to detect the vertebral boundaries that were otherwise undetectable in the CT modality. Crucially, the PET image volumes I use were obtained using the uncommon "FLT" radiotracer [5] (described in Section 2.1.2) which is actively being studied for potential clinical use in HSCT [6, 7].

The end result is a robust method of instance segmentation for the vertebral bodies of post- hematopoietic stem cell transplant patients from dual-modality CT/PET image volumes. The convolutional neural network -based semantic segmentation method I present in Chapter 3 is also able to segment the pelvis and sternum bone structures.

1.1 Motivation for This Work

This thesis is primarily motivated by ongoing research that studies cell proliferation as it occurs in the bone marrow compartments of HSCT patients by using PET imaging in combination with the uncommon FLT radiotracer [5, 6, 7]. Previously, a similar imaging analysis of marrow cavities was conducted by Agool et. al for patients with aplastic anemia [8]. PET imaging of the FLT radiotracer provides us with a non-invasive tool for examining hematopoietic stem cell (HSC) proliferation within bone marrow cavities post-transplant [5]. In contrast to the typical method of examining the body's marrow cavities (i.e., targeted invasive biopsies), the cell proliferation measurements obtained by FLT-PET imaging are an informational boon. The FLT radiotracer offers a representation of hematopoietic cellular proliferation and makes it possible to track the patterns of HSC proliferation towards engraftment throughout the entire body [7]. FLT was introduced as a PET imaging radiotracer in 1998 by Shields et. al [9] and has since been used in various preclinical trials, mostly in oncology [5].

There is significant potential for improving HSCT patient outcomes through FLT-PET imaging. Williams et. al have shown that FLT-PET imaging can provide detection of engraftment early enough to allow for the administration of a second HSCT in cases of engraftment failure [6]. In addition, a potentially life-saving research question (which is regrettably not answered in this thesis) is: can analysis of spatiotemporal cell proliferation patterns leading to engraftment be used to predict HSCT patient outcomes post-engraftment? Towards answering this question, it may help to have more granular information about stem cell proliferation activity within the body, and the bone marrow compartments within the vertebra, pelvis, and sternum are areas where cell proliferation is typically high post-transplant [6].

Presently, this post-transplant FLT-PET data is assessed in a labor-intensive process where expert physicians locate, draw, and view regions of interest (ROIs) in 2D using special medical imaging software packages. This labor requirement makes analysis of volumetric structures, such as bone, difficult or impossible in clinical practice. Automatic segmentation and visualization of 3D bone marrow cavity ROIs throughout the entire body, when combined with the (preclinical, in this setting) FLT radiotracer, would allow medical researchers or physicians to study the cell proliferation patterns of HSCT patients in finer detail; and with much less time invested.

1.2 Contributions and Organization

In this thesis I provide a system for robust automated instance segmentation of the individual vertebral bodies using joint CT and FLT-PET image volumes. The system is also able to segment the pelvis and sternum bone structures. The patient image volumes used in this thesis are under-sampled in the axial direction. The practice of under-sampling is used to limit the total radiation exposure to patients, in this case during the vulnerable period post-HSCT of recovery to engraftment [6]. The poor resolution along the axial direction makes the vertebral body instance segmentation task considerably more challenging. In many respects this thesis can be viewed as an extension to related work which used the same CT/FLT-PET volume data [10, 11]. These methods will be differentiated from my contributions in Chapter 2, along with a brief review of some relevant medical terminology.

Convolutional neural networks (or informally, *convnets*) play a central role in this thesis. I present two distinct convnets for image processing tasks: A semantic segmentation convnet for the automatic segmentation of bone structure regions of interest (Chapter 3), and a convnet for learned asymmetric super-resolution to more accurately upscale under-sampled medical images (Chapter 4). Background on these artificial neural network architectures is provided in Chapter 2. Chapter 2 also includes background on the image processing tasks of image segmentation and super-resolution, which are two topics explored as part of my original contributions in the following chapters.

1.2.1 Multi-View Ensemble 2D U-Net Model

I use CT image volumes for an initial semantic segmentation of the vertebral body, pelvis, and sternum bone structure ROIs. For this I train a semantic segmentation convnet (specifically, a modified U-Net [12]). Presently, convnets are widely used for medical image segmentation tasks, and U-Net has proven to be an effective convnet architecture when paired with data augmentation on smaller training datasets [13, 14, 15, 16]. I was fortunate to inherit ground-truth masks (heretofore unused in publication) for the vertebral body, pelvis, and sternum bone structures for a small dataset of CT image volumes [17]. I use these ground-truth annotations to train the convnet to segment the regions of interest from full body 3D image volumes. The vertebral bodies, pelvis and sternum are ROIs with high levels of hematopoietic stem cell proliferation post-HSCT, and these segmentations are used in Chapter 5 for extraction and analysis of PET data.

The current best-in-class convnets for semantic segmentation of image volume data use computationally expensive 3D convolutional layers [18, 19]. The acceleration hardware used for the convnet training and inference in this thesis is a Nvidia GTX 1070, which has 8GB of video RAM (VRAM) available for training. This computational limitation puts the 3D semantic segmentation convnets just out of reach. Instead, I use a multi-view ensemble of 2D U-Nets to increase the accuracy of the segmentation by incorporating information from more than one anatomical plane. In this ensemble method, the results of three different 2D U-Nets are combined to form a single prediction volume. Each of the three U-Nets has been trained on scans from a different "view" – in this case the views are the sagittal, axial, and coronal anatomical planes. This approach is discussed in detail in Chapter 3. The ensemble method effectively segments the vertebral body, pelvis, and sternum bone structure object classes.

1.2.2 Asymmetric Upsampling Module for Super-Resolution Convnets

Following the initial semantic segmentation, we can achieve a more granular segmentation by separating the individual vertebral bodies into distinct 3D objects (instance segmentation). The poor resolution and high level of noise along the axial direction of the CT volumes make this task particularly challenging. Many of the vertebrae above the lumbar region are impossible for even a human to differentiate visually, as is exhibited by the lack of separation between vertebra in the human-labelled ground-truth segmentation masks for the vertebral body class. While convnets have achieved excellent (and in some cases super-human) performance on computer vision tasks which humans find intuitive, we generally do not expect them to perform well on tasks that are nearly impossible for humans.

To increase the poor resolution of the CT volumes along the axial direction, I explore the use of single-image super-resolution (SR) as a preprocessing step to the semantic segmentation. For this, my working hypothesis is that SR might be able to reconstruct enough details in the spinal regions of the CT image volumes to allow individual vertebrae to be separable, by human or machine. The CT volumes of the HSCT patient dataset are under-sampled such that the voxels have an anisotropic size of approximately $1 \times 1 \times 3$ mm. I endeavor to use SR to increase the resolution in the third (axial) dimension to create inferred isotropic voxels of size $1 \times 1 \times 1$ mm by sequentially applying an SR algorithm to 2D image slices.

Convnet architectures have recently achieved state-of-the-art performance in image SR [20]. These models have focused on symmetric scaling factors where both dimensions of an image are scaled equally. Few of them have the capability to perform SR with an *asymmetric* scaling – one which resizes the dimensions of an image with a different scaling factor for each dimension. In fact, I am aware of only one example of a convnet for asymmetric scaling factors that has been reported in the literature; but it focuses on generally asymmetric scaling factors where both dimensions are scaled at different factors and does not test the case where one dimension is not scaled [21]. *Purely* asymmetric upscaling (where only a single dimension is scaled while the other is held to unity) is required for the intended use case of super-resolving the under-sampled image volumes. To address this I propose, implement, and test a "drop-in" purely asymmetric upsampling module based on transposed convolutional layers that can be used in place of the default sub-pixel [22] symmetric upsampling modules that are used in most of the state-of-the-art SR convnets [20]. This asymmetric upsampling module is discussed in detail in Chapter 4.

1.2.3 Vertebral Boundary Detection Algorithm

While the asymmetric SR convnet achieves good performance on the asymmetric SR task compared to the baseline of naïve interpolation, it does not reconstruct enough detail of the spine along the axial direction to allow segmentation of individual vertebral bodies from the under-sampled CT volume image data. However, the boundaries between the vertebral bodies that are undetectable in the under-sampled CT modality are found to be well-resolved in the under-sampled FLT-PET modality. The reason the boundaries between the vertebral bodies are visible in the FLT-PET but not CT is at least twofold. First, the FLT-PET data was captured at a slightly higher resolution in the axial dimension than the CT data. Put another way, it is "*less* under-sampled". Second, the cell proliferation activity that the FLT-PET makes visible has a smaller spatial extent than the vertebral body bone in which it resides. I develop an algorithm that enforces expected vertebrae size as an anatomical prior to search for the vertebral boundaries from FLT-PET image volume data. The method is very simple. After reducing the FLT-PET data to a single dimension along the axial direction, the algorithm iteratively locates vertebral boundaries one-at-a-time by searching the signal for the minimum within a window defined by the vertebral span prior. This approach has great performance on the HSCT patient dataset for detecting the lumbar and thoracic vertebral boundaries. The methods and the algorithm are discussed in Chapter 5.

1.2.4 3D Visualization Tool for Segmented FLT-PET ROIs

Finally, I develop some tools for generating visualizations of the FLT-PET data in the vertebral column, the individual vertebral bodies, the pelvis, and the sternum. These visualizations use translucent isosurfaces to display levels of cell proliferation activity, giving a sense of the overall proliferation patterns in 3D. The levels of the various isosurfaces can be determined programmatically from the distribution of FLT-PET intensity values or they can be set to specific target values. The visualizations have the potential to assist clinicians by providing a more comprehensive view of the FLT-PET data with less labor than current commercial software, and in the future may also facilitate fully automated assessment of HSCT recovery to engraftment. These results are presented in Chapter 5.

1.2.5 List of Specific Contributions

Below I provide a succinct list of my original contributions contained in this thesis.

- A multi-view 2D ensemble multi-class U-Net model for the semantic segmentation of the column of vertebral bodies, the pelvis, and the sternum from CT image data. The flexibility of the U-Net model to predict multiple complex unconnected bone structures is an improvement over the previous methods, which were hand-crafted to segment only the vertebral bodies.
- An asymmetric upsampling module based on transposed convolutional layers that can be easily “dropped in” to many existing state-of-the-art single-image SR convnets to enable purely asymmetric super-resolution. Asymmetric SR convnets may be useful as a preprocessing step in place of naïve interpolation methods to reconstruct under-sampled medical images with anisotropic voxels to higher-resolution isotropic volumes.
- A boundary detection algorithm that enforces a vertebra size prior to detect the boundaries between the individual vertebrae in under-sampled CT/FLT-PET volumes. When combined with the vertebral body column segmentation of Chapter 3 this allows for the instance segmentation of vertebral bodies, even in cases where a human cannot distinguish the vertebral boundaries in the CT modality.
- A tool for automatically generating 3D visualizations based on isosurfaces of the FLT-PET image volume data in segmented ROIs. The visualizations may help medical researchers analyze the FLT-PET data in greater detail.

Next, in Chapter 2, I provide the necessary background for these contributions and a review of related work.

Chapter 2
Background

In this Chapter I provide the required background to understand and motivate the medical image segmentation tasks undertaken in this thesis, starting with a review of the medical terminology. An introduction to convolutional neural networks is also provided, followed by sections dedicated to the image segmentation and super-resolution tasks. Previous works related to the image segmentation and super-resolution tasks described in this thesis are reviewed in Section 2.3 and Section 2.4, respectively.

2.1 Anatomical, Biological, and Medical Imaging Terminology

2.1.1 Hematopoietic Stem Cell Transplantation (HSCT)

Hematopoietic stem cells (HSCs) act as progenitors for all other types of blood cells in a process called definitive hematopoiesis [22]. In definitive hematopoiesis, an HSC differentiates into the various specialized blood cell and renews itself through asymmetric division. In this way the hematopoietic system produces new blood cells while maintaining the population of HSCs [23]. Hematopoiesis takes place in the bone marrow – semi-solid tissue found within cancellous ("porous") bone regions. Within the vertebrae, the cancellous bone where HSCs are found is called the vertebral body.

Hematopoietic stem cell transplantation is a medical procedure that seeks to regenerate functional bone marrow in patients by intravenous injection of HSCs [24]. It is a high-risk procedure used to treat a variety of life-threatening conditions. The most common applications for HSCT are immune deficiencies and certain malignancies occurring in the bone marrow or blood, such as leukemias [24]. In the latter cases HSCT allows patients to recover from the use of myeloablative ("high-dose") radiation and/or chemotherapy, where high-dose radiation treatments ablate a patient's bone marrow and HSCT is used regenerate it. Since the inception of HSCT, applications for the procedure have expanded to include marrow failure syndromes and congenital red cell disorders [24]. A successful *engraftment* is the primary goal of HSCT. Engraftment has occurred when the donor HSCs have proliferated to the extent that they can self-sustain long term hematopoiesis within the body's marrow compartments (a threshold for which there are various practical definitions) [25].

2.1.2 3'-Deoxy-3'-[^{18}F]Fluorothymidine ([^{18}F]FLT)

[^{18}F]FLT, or in this thesis simply FLT, is a radiotracer used in conjunction with positron emission tomography (PET) to measure hematopoietic cell proliferation non-invasively [9]. Cell proliferation is the process by which a cell grows and divides, producing two daughter cells. In some preclinical trials for cancer research FLT has been used as an alternative to the more common radiotracer 2-[^{18}F]-fluoro-2-deoxy-D-glucose ([^{18}F]FDG), which measures the cellular metabolism in the form of glucose uptake [5]. In contrast to FDG, FLT offers a granular, more exacting view of hematopoietic cell-proliferation and does not accumulate in metabolically active organs [5]. Both radiotracers, FLT and FDG, can be measured by PET and then normalized to units of standardized uptake value (SUV). SUV is the ratio of the radiotracer activity per unit volume to the average radiotracer activity of the entire body [26]. The calculation of SUV is non-trivial and is susceptible to various sources of error [27]. Depending on what radiotracer is used (i.e., FLT, FDG, or another radiotracer), SUV represents the activity level of different cellular mechanisms. In the medical research literature SUV is commonly associated with FDG, but that is not the case in this thesis, where the FLT radiotracer is used to measure hematopoietic cell proliferation [6]. For this thesis, SUV normalization of the FLT-PET scans of the patient dataset has been already determined by the work of Williams et. al [6], and the "FLT-PET" volumes are already in units of SUV (i.e., each voxel is valued according to the SUV of the FLT radiotracer for the anatomical volume it represents).

2.1.3 Anatomical Planes

In this thesis the terms *axial*, *sagittal*, and *coronal* are used to describe the standard anatomical planes that bisect a human body. The axial plane bisects a human between the head and the toes, the sagittal plane between the eyes, and the coronal plane bisects the "front" from the "back". Examples of the three planes are shown in Figure 1. Sometimes I will refer to the axial, sagittal, or coronal "axis" or "direction". This should be taken to mean "the axis normal to the anatomical plane" and is meant to spare myself and the reader from having to keep track of an additional handful of terms used to describe anatomical direction that are more descriptive than required for this thesis.

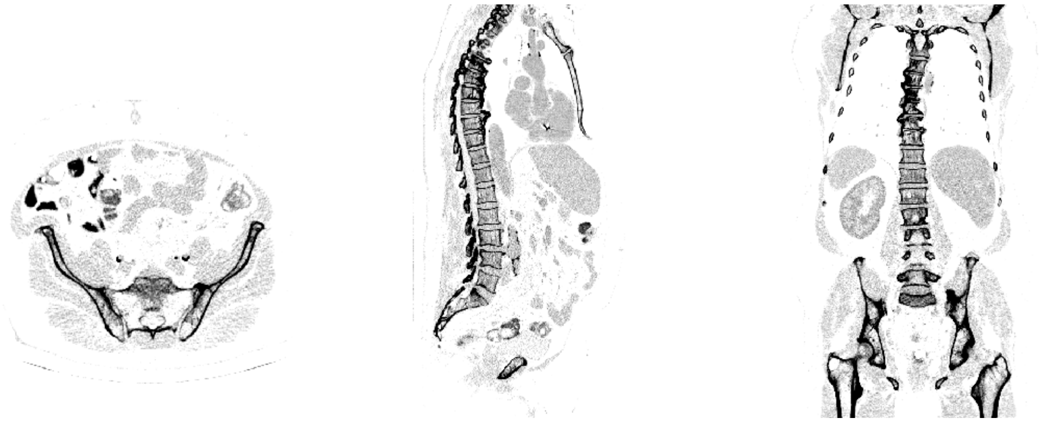


Figure 1. From left to right, CT scans from the axial, sagittal, and coronal anatomical planes. Images sliced from the VerSe 2019 dataset [28, 29, 30].

2.1.4 The Spine and Vertebrae

Image segmentation of the spine plays a central role in this thesis. The bottom-most five vertebrae in the spine are referred to as *lumbar* vertebrae and are labelled upwards starting above the coccyx ("tailbone") as: L5, L4, …, L1. Following the lumbar vertebrae are the twelve *thoracic* vertebrae labelled upwards as: T12, T11, …, T1. Last are the *cervical* vertebrae, of which there are seven, similarly labelled upwards as: C7, C6, …, C1. The size, shape, and even structure of each vertebra varies. Figure 2 shows an example of a lumbar vertebra. This thesis is concerned mostly with the segmentation of the *vertebral bodies*; these contain cancellous bone tissue and, in healthy living vertebrates, the bone marrow [6].

2.2 Artificial and Convolutional Neural Networks

Recently the deep convolutional neural network (or commonly, convnet) has achieved state-of-the-art performance on many computer vision tasks that can be formulated in terms of "spatial pattern recognition" – tasks like object detection [31] and image segmentation [2]. Convnets are a specialized deep artificial neural network that incorporates convolutional layers. A comprehensive introduction to the deep artificial neural network is provided by Goodfellow et. al [32], but here I will focus on the concepts most relevant to the convolutional neural networks used in this thesis.

Artificial neural networks (ANNs) are often used for supervised machine learning. In supervised learning a model is trained by presenting a training set containing examples of inputs and their associated outputs (or targets) to the learning algorithm [32]. When

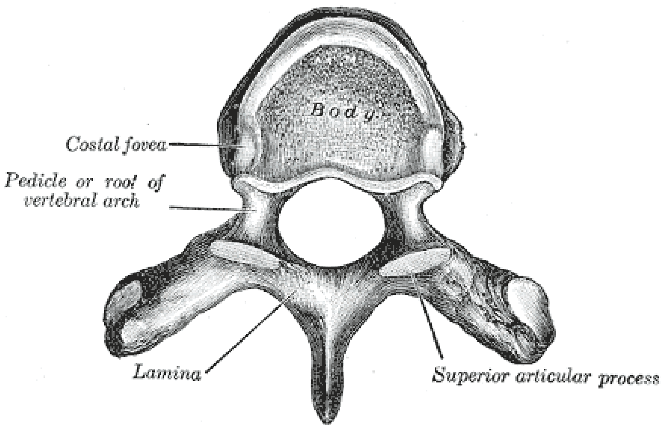


Figure 2. An example lumbar vertebra. The vertebral body is labelled “Body”. Illustration from “Anatomy of the Human Body” [33].

“training” a supervised ANN, the algorithm iteratively makes predictions (or inferences) which are graded against the target by a performance-measuring loss (or cost) function, and the measured loss is used to update the numerical parameters that the learning algorithm uses to make predictions. The goal of the training process is the reduce the generalization error (the error the model yields on unseen data) by minimizing the training error [32]. During training, the generalization error is often tracked by intermittently inferencing a validation set that is unseen by the training algorithm. A test set can be held in reserve to test the final model on unseen data.

2.2.1 Deep Artificial Neural Networks

A deep artificial neural network is a multi-layered computational structure for machine learning that iteratively and incrementally learns representations (or features) from a distribution of inputs and maps these features to an output. The fundamental unit of computation in an ANN is the artificial neuron, the function of which can be considered a weak analogue to that of the biological neuron [32]. In the most typical case the artificial neuron ingests an input vector, computes an inner product with the learned parameters (also commonly called weights), applies a learned bias and nonlinear activation function, and outputs the result. Formally, the output h of an artificial neuron with input vector $\boldsymbol{x}$**,** nonlinear activation function ϕ, weights vector $\boldsymbol{w}$ and bias b is given by [32]

$$h(\boldsymbol{x}) = \phi(\boldsymbol{w}^T \boldsymbol{x} + \ b). \tag{1}$$

The neurons of an ANN are most often grouped together in N-dimensional interconnected layers, and layers between the input and output layers of a neural network are called hidden layers [32]. The number of hidden layers is often referred to as the “depth” of the network. The state-of-the-art artificial neural networks being developed presently typically have many hidden layers (e.g., ResNet-152 with up to 152 hidden layers [34], RCAN with over 400 hidden layers [35]). These layers can be connected to each other’s neurons in different ways and by different operations, giving rise to a taxonomy of different types of layers and connections [36]. Not all layers contain learnable parameters. Pooling layers are an example of a layer with zero learnable parameters [37]. A pooling layer reduces the dimensionality of the preceding layer by performing an operation (average, max, etc.) on subsets of the preceding layer’s outputs. A particular configuration of layers and the operations which connect them is commonly referred to as a deep neural network architecture [32].

It is the incorporation of the nonlinear activation function ϕ that allows the neural network to represent complex nonlinear functions [32]. Historically, smooth and saturating activation functions such as the sigmoid and hyperbolic tangent have been used for the activation of an ANN’s neurons. However, presently the rectified linear unit (ReLU) and its variants are the recommended practice [38]. The basic ReLU is a simple piecewise function that returns 0 for negative arguments and returns identity for positive arguments [39].

2.2.2 Loss, Optimization, and Backpropagation

The ability of a modern neural network to learn effective representations, or features, is enabled by gradient-based optimization. Optimization in ANNs involves iteratively minimizing a loss function (also commonly called a cost or objective function) [32]. The choice of loss function will depend on the task [40, 41]. It is common that the mathematical representation of performance on the true task is not suited for iterative-based optimization methods like those used for training neural networks [32]. The goal in such cases is to find a suitable proxy for the true loss whereby minimizing the proxy function generally increases the performance on the true task. Some common loss functions (often used as proxies) include L1 loss, L2 loss, log loss (also called cross entropy loss), and hinge loss [40]. There are a multitude of specialized loss functions found in the neural network literature. The loss functions used in this thesis will be introduced as they arise so that they may be better understood with more context.

Optimization has been an active area of research alongside the recent deep learning boom. The focus of this research has been to get models to converge more consistently, faster, and towards lower minima during training [42, 43, 44, 45]. Still, nearly all of these recent optimization algorithms for modern deep ANNs use some form of stochastic gradient descent (SGD) [46]. SGD is an application of "vanilla" gradient descent, the latter being a method where a convex cost function can be optimized by procedurally following the direction opposite its gradient towards a global minimum [47]. We can define the gradient descent optimization procedure for a deep neural network as trying to find the learnable parameters $\boldsymbol{\theta}$ (also known as the weights and biases) that minimize a particular loss function $L(\boldsymbol{\theta})$ over the available training set. In deep neural networks the dimensionality of $\boldsymbol{\theta}$ is often so huge that computing the gradient over the entirety of the training set at once is computationally prohibitive [32]. This is where SGD comes in, which considers the gradient $\nabla_\theta L(\boldsymbol{\theta})$ to be a statistical expectation that can be estimated by iteratively taking a single training pair (or a random sampling of pairs called a *minibatch*), computing the gradient of the loss function with respect to the parameters $\boldsymbol{\theta}$, and updating the parameters $\boldsymbol{\theta}$ in the direction of steepest descent (opposite the gradient) [46]. The only question left to answer is: "how fast do we descend?", and the answer to this question is the learning rate. The SGD parameter update process just described can be succinctly described by (adapted from [32])

$$\boldsymbol{\theta} = \boldsymbol{\theta} - r\nabla_\theta L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \tag{2}$$

for a single-example-per-iteration case of a training pair of input $\boldsymbol{x}$, target $\boldsymbol{y}$, and learning rate r. Or, for the minibatch case by [32]

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \frac{r}{n}\sum_{i=1}^{n} \nabla_\theta L(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta})\,, \tag{3}$$

where n is the minibatch size. The parameters $\boldsymbol{\theta}$ are iteratively updated in this way until the ANN converges to solution that performs well enough for the intended application. Selecting the best learning rate is a subject that has recently received significant attention, and schemes or schedules have been designed to adjust the learning rate algorithmically [48, 49, 50]. A classic example of such a method is momentum, where successive parameter updates in a similar direction increase the learning rate (with an exponential decay term to control the growth) [51]. More recently, adaptive learning rate algorithms (many of which incorporate momentum as a feature) have gained popularity [42, 43, 44, 45]. Adaptive learning rate algorithms adjust each parameter's learning rate individually during the training process. A widely used adaptive learning rate optimization algorithm is the Adam optimizer [44]. A recent extension of Adam, AdamW, changes the weight decay component of Adam from additive to multiplicative, which has shown improved generalization performance for some tasks [45]. In this thesis I use both Adam and AdamW for training the convnet models.

The optimization algorithms just described are fundamentally SGD. As such, all rely on gradients of the loss function with respect to the parameters, as shown in (3). In deep ANNs the computation of these gradients is almost always accomplished using the backpropagation (colloquially "backprop") algorithm [52, 32]. Backprop is used for automatic differentiation, and it is at its core a computationally efficient application of the chain rule of calculus.

2.2.3 Convolutional Neural Networks

The convolutional layer is what differentiates a convolutional neural network from the more general ANN. The convolutional layer has learned parameters $\boldsymbol{\theta}$, but the parameters here are unique in that the weights explicitly form filters or "kernels" used in convolution operations on the output of the previous layer [32]. The convolution operation ensures neurons are only connected to local neighborhoods of neurons in the previous or following layers; those neighborhoods are limited by the kernel size, which is typically much smaller than the image size. This makes them very efficient spatial feature extractors, particularly when compared to the fully-connected layer where each output neuron connects to every input neuron [32].

Ironically, the convolutional layers implemented by many machine learning frameworks use cross-correlation instead of the convolution operation [53, 54]. Cross-correlation can be viewed as convolution without flipping the kernel. Since the parameters of the kernel are learned *and* cross-correlation is just convolution with a mirror-image kernel, the two methods are effectively equivalent in that they have the same capacity for learning spatial representations [32]. Practically, the cross-correlation operation looks like a "sliding window filter" where a filter/kernel slides around an input and computes sums of element-wise products at each new location, mapping the result to a feature map in the next layer. Convolutional layers often perform many such operations in parallel, allowing them to expand or reduce the feature space by increasing or decreasing the number of feature channels at the layer's output (e.g., in a U-Net the initial "encoding" arm performs expansion of the feature space, while the "decoding" arm performs reduction of the feature space [12]). When an N-dimensional convolutional layer sees $(N+1)$-dimensional data at its input, the kernels become $(N+1)$-dimensional "stacks" of learnable N-dimensional filters, where each stack is associated with its own output feature map. The typical design parameters for a convolutional layer are [53]:

- **Number of input feature channels**: Determines the "width" of the stacks of $(N+1)$-dimensional kernels used for the N-dimensional convolution on the previous layer's outputs (or, commonly for images, "feature maps").
- **Number of output feature channels:** Determines the total number of $(N+1)$-dimensional kernels used for N-dimensional convolutions on the previous layer's output feature maps. Each $(N+1)$-dimensional kernel maps to an output feature map.
- **Kernel size:** For an N-dimensional convolution, the kernel size is an N-dimensional parameter that determines the neighborhood size of the filter kernel. Note that the actual kernels used in the convolution operation have an additional dimension with size equal to the number of input feature channels.
- **Stride**: Outputs from the previous layer can be downsampled during the convolution operation by adjusting the "stride". Visually, the stride parameter controls the step size of the sliding kernel. Practically, increasing the stride of the convolutional layer reduces the size of the next layer. In an N-dimensional convolutional layer, the stride parameter is N-dimensional (i.e., the stride can be set individually for each dimension of convolution operation).
- **Padding:** Padding edges of the input data is sometimes required to maintain or adjust the size of data output from a convolutional layer.

2.2.4 Batch Normalization

Batch normalization was proposed by Ioffe et. al as a means to improve the training of a deep artificial neural network by reducing "internal covariate shift" [55]. By applying batch normalization to outputs of learnable neural network layers, training the neural network becomes faster for some tasks. Essentially, batch normalization computes the mean and variance at each dimension of a layer's outputs and uses these statistics to normalize the values of the outputs [55]. Today, batch normalization layers are provided as modules for the popular deep learning frameworks [53, 54] and have been used alongside convolutional layers in image processing tasks like super-resolution [56] and object segmentation [18].

2.2.5 Data Augmentation

Data augmentation is an often-used technique in supervised machine learning in general and is prevalent in convolutional neural networks for image processing tasks [57]. The goal of data augmentation is to increase the diversity of the training set. It can be viewed as a form of regularization in that it is used to reduce generalization error but may increase training error [58, 32]. By performing various realistic transformations on the available training data, the network learns features that better generalize on those transformations. There is a critical guiding rule for data augmentation, which is that the transformations should generate reasonable examples from the distribution of images from which a training set is sampled [32]. Reasonable geometric transformations for image data often include mirroring, rotation, scaling, and shifting of the image. Other augmentations operate on the pixel data; these may include histogram/contrast manipulation, adding noise, adding occlusions, or color manipulation [57]. When image augmentation is implemented, augmentations are typically used in a stochastic fashion where the chance of application or parameters of augmentation are randomly determined.

2.3 Image Segmentation

Image segmentation is a digital image processing or computer vision task where the goal is to classify, label, or otherwise partition parts of an image from each other and the background [59]. For the segmentation of objects, one typically wants to achieve one of two related goals – semantic segmentation or instance segmentation [2, 60]. In semantic segmentation the goal is simply to classify all the pixels constituting the objects of interest. In instance segmentation the goal is to classify all the pixels constituting the objects of interest *and* to represent each individual object as a separate instance of that object class. When objects of the same class do not overlap in an image and are not occluded by objects of other classes, instance segmentation is a trivial extension of a semantic segmentation algorithm (e.g., by application of a connected component algorithm to the semantic prediction map). However, when multiple objects of the same class overlap, touch, or otherwise appear contiguous, instance segmentation becomes a challenging task itself [61].

The primary aim of this thesis is one of instance segmentation: to segment and label each individual vertebral body from dual-modality CT/FLT-PET image volumes. While instance segmentation is the end goal, semantic segmentation is a critical component in the segmentation framework I develop. The process starts with a semantic segmentation convnet where the voxels of vertebral bodies are classified as “vertebral body voxels” with no distinction between them. The under-sampled axial CT scans make the vertebral bodies difficult to instance algorithmically, as the individual bodies appear to be connected when viewed in the sagittal and coronal views. This compels me to search for other methods to label the vertebral bodies into separate instances, and ultimately label them with their anatomical names (L1, L2…, T1, T2…, etc.).

2.3.1 Traditional Image Segmentation

Prior to the recent insurgence of the convolutional neural network, the image segmentation task was accomplished with a variety (and often a combination) of hand-tuned algorithms [59, 62, 63, 64]. Today these methods are sometimes used in conjunction with the more recent convnet-based segmentation methods [65].

Edge-based methods aim to locate the edges of objects in an image and use them for the segmentation task. Most edge detection algorithms rely on spatial filter kernels to measure the local gradients of the pixels in an image [59]. More advanced algorithms like the canonical Canny Edge Detector have improved edge detection by qualifying edges with mathematical criteria which reduce false positive and false negative edge responses, optimize for edge localization accuracy, and impose an "edges have single-pixel width" rule [66]. For an object segmentation task, the edge response of the object must still be differentiated from the other edge responses in the global image.

Threshold-based methods partition objects in images based on their pixel intensity values. Simple "global" thresholding (where every pixel in the image is classified based on a single threshold value) is an effective method when there is high and consistent contrast between the object(s) and the background. In cases where contrast between object and background changes throughout an image, more advanced adaptive thresholding techniques may be successful [59]. There are many situations where even advanced thresholding-based methods are not viable. As a relevant example, consider the vertebrae segmentation task that is the focus of this thesis. Thresholding can be an effective method for segmenting bone vs. not bone; the 3D adaptive method presented by Zhang et al. [67] is suited for this kind of task. However, as shown in [67], even adaptive thresholding techniques are not able to effectively differentiate between different cortical bone structures.

Region-based methods attempt to find regions by algorithmic approaches with an emphasis on spatial relationships and feature similarity between neighboring or nearby pixels. A common region-based method is region growing. Region growing methods rely on the initialization of one or more seed pixels from which the segmentation is "grown" by iteratively comparing neighboring pixels against decision criteria [59]. A decision criterion can be a simple threshold, whereby the region growing algorithm is effectively a connected component extraction. Region-based methods are often used in combination with thresholding-based methods. This is the case for the previously mentioned 3D adaptive thresholding method [67] which incorporates a region growing algorithm to isolate individual bone structures from the thresholded full-body CT scans. Other region-based methods include segmentation via k-means clustering [68], super-pixels [69], graph cuts [70], and morphological watersheds [59].

Model-based methods use prior information about the expected shapes of the objects in an image and use this to search for and label the objects. The most basic implementation of a model-based image segmentation algorithm may be object extraction via rigid template matching [71]. This kind of rigid model works well in applications where objects are highly regular in shape (e.g. in vision tasks for manufacturing). In contrast, anatomical components in humans can be non-rigid and are decidedly non-uniform; the exact shape of anatomical structures varies across our species' population [72]. To apply model-based segmentation techniques to medical images, flexible models have been developed based on the *statistics* of *shape,* where statistical models are derived by machine learning on ground-truth segmentation data [73]. These *statistical shape models* (SSMs) have been successfully deployed to solve a variety of medical image segmentation tasks (including spine and vertebrae segmentation); an overview of the technique and a survey of SSM implementations for 3D medical image segmentation are contained in a review article by Heimann and Meinzer [73]. A more recent and particularly relevant work by Neubert et al. shows a statistical shape model for vertebral body segmentation of MR volume images achieving a dice score of 0.91 [74]. Note that so-called "atlas-based" methods such as [75] are essentially model-based methods where segmentation is performed by registering templates or "atlases" to objects in an image; this latter work achieved a mean Dice score of 0.94 across their test dataset.

Combinations of these approaches were prolific in medical image segmentation before the deep-learning-based convnet methods became the state-of-the-art. An excellent survey article comparing the performance of several such traditional methods in the vertebrae segmentation task is provided by Yao et al. [76]. This paper reviews algorithms submitted to the "Vertebrae Segmentation Challenge" held during the 2014 International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI). These algorithms, all of which use some form of shape model, report segmentation Dice scores ranging from 0.868 to 0.947. The authors also show that the segmentation task becomes harder when segmenting vertebrae higher up the spine; these methods reported a mean Dice score of 0.933 on the lumbar (lower) vertebrae, compared to a mean Dice score of 0.867 on the upper thoracic vertebrae. It should be noted that this work focused on the segmentation of *entire* vertebrae and not just the vertebral bodies.

2.3.2 Vertebral Body Segmentation for HSCT Patients

This thesis is largely motivated by recent work that studies cell proliferation within the bone marrow cavities of post-HSCT patients by PET imaging of the unique FLT radiotracer [6, 7, 10, 11]. Specifically, I use the same dataset as Williams et al. in [6]. Their pilot study showed that dual-modality CT/FLT-PET imaging can be used to measure cell proliferation throughout the entirety of the body's bone marrow compartments non-invasively. The authors suggest that FLT could be used as a biomarker for hematopoietic recovery [6]. The dataset used in these works and in this thesis consists of joint CT/FLT-PET scans of hemopoietic stem cell transplant patients. All patients underwent myeloablation to obliterate their existing bone marrow including hematological malignancies, followed by hematopoietic stem cell transplantation. The patients were imaged on multiple days during their treatment: on the day before stem cell transplantation (when the patient's haemopoietic system has been ablated), between 5 and 8 days after transplantation, and on the 28th day after transplantation [6, 10, 11].

The analysis of post-HSCT cell proliferation in the pilot study [6] was enabled in part by the automated marrow cavity segmentation methods from Nguyen et al. in [10] and [11]. Their vertebral body segmentation framework uses a bilateral filter as a preprocessing step, a graph-cut method to perform the segmentation of the full body bone structure, an iterative thresholding algorithm to isolate the column of vertebral bodies, a Kalman Filter algorithm to locate the boundaries between the vertebrae, and a morphological erosion filter to shave cortical bone from the cancellous bone target. Although I use the same patient data in my experiments, I accomplish the vertebral body segmentation using entirely different methods. The authors in [10, 11] grade their algorithm using what they call a "percent agreement" criterion:

$$C_{PA}(V_P, V_{GT}) = 100 \times \frac{|V_P \cap V_{GT}|}{|V_{GT}|} \ , \tag{4}$$

$$= 100 \times \frac{TP}{TP + FN} \ , \tag{5}$$

where V_P represents the predicted segmentation volume, V_{GT} represents the ground-truth segmentation volume, and the notation $|X|$ represents taking the number of positively identified class voxels contained in the binary segmentation volume X. Equation (5) is written in the language of a confusion matrix (see Table 1 for definitions of the terms TP and FN). This metric only measures the ratio of true positive voxels to ground-truth voxels. Another name for this criterion is the "true positive rate", or TPR. The problem with relying on this metric as the sole measure of segmentation performance is that it ignores false positives. The blind spot this can create is clearly seen by application of a test case where we declare "the entire prediction volume V_P is bone". In such a case $|V_P \cap V_{GT}|$ is equal to $|V_{GT}|$ and the "percent agreement" (or TPR) criterion C_{PA} reports 100% accuracy even though many (and possibly most) of the V_P voxels are false positives. For this reason, I disagree with the authors' claim that "a perfect segmentation result is 100%" [10] under this metric. A general-purpose performance metric for image segmentation should account for cases of pixel misclassification along with the true positives. Thus, in my experiments I use more representative metrics to grade the vertebral body segmentation task. In [10] the authors report a mean C_{PA} (or true positive rate) on the test set of approximately 91% for the vertebral body bone structure.

2.3.3 Performance Metrics for Image Segmenation

Heretofore I have mentioned a few performance metrics for the image segmentation task. There are some widely reported simple-but-effective ways to grade an image segmentation [13, 14, 15, 16]. These are easily computed from elements of the canonical binary confusion matrix, shown in Table 1, which provides a frequency-based representation of the classification ability of an algorithm [77]. At its core, image segmentation is a classification task. One of the more naïve performance metrics for image segmentation is the pixel-wise accuracy metric, or pixel accuracy. This metric simply answers: "what proportion of the pixels in the image were classified correctly?" and can be mathematically described (in terms of the elements of the confusion matrix) by [77]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \ . \tag{6}$$

Accuracy makes intuitive sense as a general-purpose classification performance metric; it weighs all the components of the confusion matrix. Accuracy ranges from zero to one. A larger proportion of correct predictions makes the accuracy go up, and a larger proportion of incorrect predictions makes the accuracy go down. Accuracy is a good

TABLE 1. THE CONFUSION MATRIX

	Actual Positives	**Actual** Negatives
Predicted Positives	True Positives (TP)	False Positives (FP)
Predicted Negatives	False Negatives (FN)	True Negatives (TN)

enough metric for many classification tasks, but it can misrepresent the effectiveness of a classification algorithm on imbalanced datasets with a small proportion of actual positive examples. Such is the case for many image segmentation tasks, where the objects are represented by a small subset of the total pixels in an image. In a hypothetical segmentation task where the objects make up 5% of an image, an "empty" prediction where all pixels are classified as background would still score 95% on the accuracy metric. So, while pixel accuracy does weigh true positive and true negative results against both types of classification errors, its descriptive ability is highly sensitive to the frequency (or infrequency) of classes appearing in a dataset. This fact motivates the use of other performance metrics for grading image segmentation.

The Sørensen–Dice coefficient (commonly known as the Dice score) and the Jaccard index (also known as intersection-over-union, or IoU) are two closely related similarity measures which mitigate the major pitfall of the accuracy metric by ignoring true negatives [78]. For a binary classification task these metrics can be defined as

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \ , \tag{7}$$

$$IoU = \frac{TP}{TP + FP + FN} \ , \tag{8}$$

where the Dice score simply weighs true positive results twice as heavily in both the numerator and denominator. Like accuracy, Dice and IoU range from zero to one. The Dice and IoU measures are positively correlated, and in fact one can calculate either metric from the other by the following relation [78]:

$$IoU = \frac{Dice}{2 - Dice} \ . \tag{9}$$

When presented with the same hypothetical image segmentation that I used to show the deficiency of the accuracy metric, Dice and IoU both report scores of 0.0 (since there were no true positives detected, and the metrics do not weigh true negatives). This result would often be considered more representative than the 0.95 reported by the accuracy metric. In many segmentation tasks we intuitively (and practically) value true positives more than true negatives. Most recent works in semantic segmentation report either Dice score or IoU, as shown in the review article from Garcia et al. [79]. Although Dice and IoU are often more representative, they have their own blind spot: without additional information they cannot tell us anything about the ability of a model to detect true negative cases.

Two other important classification metrics are *sensitivity* and *specificity.* These are perhaps best explained by their respective alternative names, the true positive rate (TPR) and the true negative rate (TNR) given by [77]

$$TPR = \frac{TP}{TP + FN} \ , \tag{10}$$

$$TNR = \frac{TN}{TN + FP} \ , \tag{11}$$

where TPR represents the ability of a classification algorithm detect a positive case and TNR represents ability to detect a negative case (the latter being related to the false positive rate by $FPR = 1 - TNR$) [77]. A convnet-based image segmentation algorithm commonly produces a prediction image where each pixel is assigned a confidence level of belonging to a certain object class [12, 80, 81]. These confidence levels are output by the network in the range [0, 1] and the threshold that decides *object* vs. *not object* may be set by the practitioner or by the algorithm itself. By decreasing this classification threshold, the algorithm can be compelled to generate more true positives at the expense of increasing false positives. This effect is perhaps most clearly seen using a receiver operating characteristic (ROC) curve which is a plot of TPR vs. FPR acquired by varying the classification threshold [77]. Such a representation is useful for selecting between models and classification thresholds in cases where true positives and false positives are valued differently.

Finally, sometimes in the medical image segmentation literature we encounter less-common segmentation metrics. When used to grade image segmentations, these less-common measures are typically provided alongside IoU or Dice. One example is the Hausdorff Distance [82], which is used to quantify the deviation of the contour (or surface, in the 3D case) of a segmentation from that of the ground-truth object mask. Hausdorff Distance and the other metrics used for 3D medical image segmentation have been reviewed by Taha and Hanbury in [78]. The article also examines the representation capabilities of the more common segmentation performance metrics discussed above.

2.3.4 Convnets for Medical Image Segmentation

Recently the convolutional neural network has become the state-of-the-art tool of choice for image segmentation tasks, including medical image segmentation [83]. Long et al. [84] were the first to train a "fully-convolutional network" end-to-end for segmentation tasks. In their work the fully-connected layers traditionally used in previous artificial neural networks for image segmentation were replaced with convolutional layers. The resulting deep learning network achieved state-of-the-art segmentation performance and large efficiency gains due to the removal of the fully-connected layers.

Building on the fully-convolutional network, Ronneberger et al. [12] introduced a deep convnet for 2D medical image segmentation called the U-Net. The name is derived from the shape of the network when drawn as a diagram, as shown in Figure 3. The U-Net has an autoencoder structure that consists of a feature encoding path and a feature decoding path. The encoding path is composed of repeated feature-expanding convolutional layers and dimension-reducing max-pooling layers. The feature-decoding path is composed of repeated dimension-expanding transposed-convolution layers ("up-conv" in Figure 3) and feature-reducing convolution operations. They also critically add skip connections to concatenate the features from the encoding path's convolutional layers to the equidimensional layers in the contracting path. This architecture allows both context and localization information to pass through the network and consequently the U-Net performs very well in image segmentation tasks [12]. U-Net has been modified and implemented in a myriad of medical image segmentation tasks, including bone [13], brain tumor [14], liver [15], lung nodule [15], and cell nuclei [16], to name just a few. U-Net has also been used for image segmentation tasks in other fields: for defect detection in pavement [85], road and building extraction from satellite imagery [86, 87], and even more recently for segmentation of roots in soil [88].

Motivated by 3D medical image data, a natural extension of convnets for 2D semantic image segmentation are convnets for 3D semantic image segmentation. One of the first such works is the 3D U-Net from Çiçek et al. who replace the 2D convolutional layers in U-Net with 3D convolutional layers [18]. They also make a few other modifications such as the incorporation of batch normalization layers into the network architecture. That same year, V-Net was developed as another 3D extension of the U-Net architecture [19]. V-Net makes more modifications to the U-Net inspired

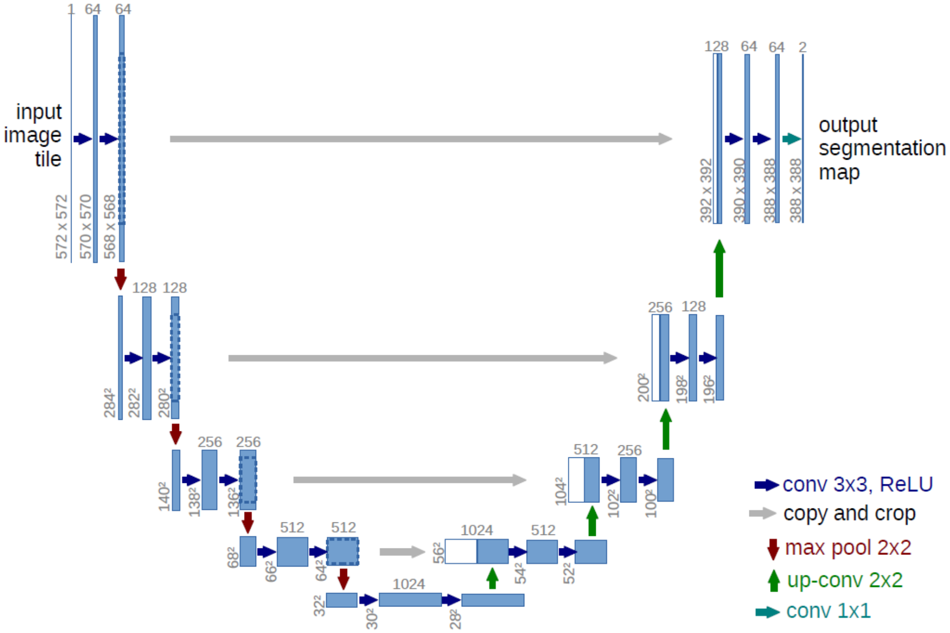


Figure 3. The original U-Net architecture. Image from Ronneberger et. al [12].

architecture. Two of the most impactful are the residual connections around groups of convolutional layers and the learned downsampling convolutional layers in place of the max-pooling layers. The challenge of these implementations is the model size and the computational complexity of the 3D convolution operations. This can be mitigated by training on volume "tiles" or "patches" sampled from 3D image data, which is the approach taken by [18]. However, training in this manner forces the network to learn representations from smaller neighborhoods, sacrificing more global information.

Seeking more computationally feasible networks that can still leverage 3D data in semantic segmentation convnets, some "pseudo-3D" semantic segmentation convnet models have been investigated. One method uses 2D convolutions on stacks of neighboring 2D inputs [89]. Instead of processing a single 2D image this network processes a stack of 2D slices as a single input with multiple channels, obtaining relevant (albeit limited) 3D context information from neighboring 2D slices. Another pseudo-3D method uses multiple 2D segmentation convnet models trained from multiple views of the input data volume [90, 91]. By learning to perform the segmentation on multiple views, these models incorporate 3D information into their segmentation predictions with much less computational overhead than what is required for a fully-3D convolutional network. Lastly, the long-short-term-memory (LSTM) variant of the recurrent neural network can be used to treat the 3D segmentation problem as a sequence of 2D segmentations [92]. These pseudo-3D implementations were developed for processing 3D medical image data.

It is difficult to compare the relative efficacy of the methods because they were developed and trained with niche datasets for a very specific research purpose.

Particularly relevant to my work is that of Shigeta et al. [91] who use two of the pseudo-3D methods just described (depth-as-channel inputs and multi-view ensemble models) to perform spinal segmentation. They report a mean Dice score of 0.964 from a model trained and validated on their in-house spinal segmentation test set – a very good result. An interesting "iterative" approach to instance segmentation of individual vertebrae via convnets is provided by Lessmann et al. [93], but it relies on the separation of the vertebrae in the sagittal views of the image volume which is lacking in the under-sampled volume data that is the focus of my work reported in this thesis. The Large-Scale Vertebrae Segmentation Challenge (VerSe) [28] was an organized semantic segmentation research competition that ran in conjunction with MICCAI 2019 and 2020. It serves as perhaps the best example of the dominance of convnet approaches in the spine segmentation task, with 23 of the 24 participating teams using a convnet-based method, the majority of those being some variety of U-Net, pseudo-3D U-Net, or 3D U-Net/V-Net. The VerSe competition reveals that in the spinal segmentation task 3D convnets perform better than 2D convnets. The highest scoring spine segmentation in the VerSe competition was a 3D U-Net with a segmentation Dice Score of 0.917. This result, when viewed against the Dice score of 0.964 reported by [91], indicate that the dataset upon which the metric is being trained, tested, and reported may have a large impact on the perceived "performance" of a model. The VerSe training and test sets are particularly diverse (at least for medical image data), with CT images captured from a variety of scanners and imaging protocols. These dataset inconsistencies make it hard to make definitive comparisons between models trained and tested on different datasets. Notably, the spinal segmentation task undertaken by these related works is different than the vertebral body segmentation task I undertake in Chapter 3. Still, these results provide a good baseline for what is possible using a pseudo-3D U-Net or fully-3D U-Net for the segmentation of complex 3D shapes.

2.4 Image Super-Resolution

Image super-resolution (SR) is the task of reconstructing higher-resolution digital images from lower-resolution digital images [20]. The problem is challenging because it is ill-posed; any given low-resolution image could have been sampled from a multitude of different high-resolution images. The goal of a super-resolution algorithm, then, is to limit the solution space as much as possible towards "good" reconstructions. Recently the deep convolutional neural network (or commonly, convnet) has delivered state-of-the-art performance on image SR [20]. The highest-performing traditional and modern convnet-based super resolution algorithms accomplish this through learned-by-example training strategies [20, 94, 95]. The common idea in these implementations is to learn generalizable representations of natural texture from local image patches for the eventual high-resolution reconstruction.

2.4.1 Traditional Image Super-Resolution

Prior to modern convnet methods, state-of-the-art SR was still machine-learning-based. One of the leading methods of this pre-convnet era is Anchored Neighborhood Regression [95, 96] which borrows from the Sparse Coding [94] and Neighbor Embedding [97] methods. These are all so-called "dictionary-based" SR methods, which use a dictionary (essentially a lookup table) to map low-resolution input patches to high-resolution output patches. Neighbor Embedding gives the dictionary more flexibility by allowing a low-resolution input patch to be approximated by a linear combination of the patches nearest to it in the dictionary (its *neighbors*), and likewise for the super-resolved output patches. Sparse Coding takes this a step further. Instead of explicitly using a dictionary of image patches, the Sparse Coding method involves learning an efficient and sparse encoding for the low- and high-resolution image spaces. Anchored Neighborhood Regression uses smaller portions of the dictionary (*neighborhoods*) in the regression problem and pre-calculates a transformation matrix for each neighborhood to lower the computational complexity at run time.

2.4.2 Convnets for Image Super-Resolution

The dictionary-based methods are not so dissimilar to the modern convnet-based methods. The conceptual analogues were pointed out by Dong et al. when they produced the first convnet for super-resolution: SRCNN [98]. SRCNN has only two hidden convolutional layers, making it a very shallow network by modern standards. It relies on bicubic interpolation as an upsampling step, where the low-resolution input image is first upsampled before its features are extracted and decoded by the convolutional layers. This simple design achieved state-of-the-art super-resolution performance, beating all the dictionary-based methods. Ironically, the authors found that deeper networks were harder to train and had little performance benefit; but since the release of SRCNN the convnet-based methods have continuously achieved greater reconstruction performance by going deeper and deeper. The authors were correct though: deeper networks *are* harder to train, in general. The increased depth of the more recent SR convnet architectures relies on the residual connection [34]. The residual connection eases the training of deep neural networks by allowing information to pass *around* layers instead of forcing information to go *through*. This frees the parameters from learning identity mappings and allows them to focus on more representative features [99].

One of the first SR convnets to implement the residual connection was SRResNet from Ledig et al. [56]. In contrast to SRCCN's shallow network of two hidden layers, SRResNet uses 16 residual blocks (or ResBlocks), where a residual connection surrounds two convolutional layers, each followed by a batch-normalization layer, for a total of 32 convolutional layers. SRResNet also uses "upsampling post- feature extraction" as opposed to SRCNN's "upsampling pre- feature extraction", allowing the features to be learned in the less computationally expensive low-resolution space. The in-network upsampling method implemented by SRResNet is a "sub-pixel" convolutional layer with learnable parameters which allow the network to learn the mapping from the low-resolution (LR) space to the high-resolution (HR) space. The sub-pixel layer was introduced by Shi et al. to reduce the computational complexity of learned upsampling that was traditionally accomplished by use of transposed convolutional layers [100].

Ledig et al. used SRResNet as the generative arm of their SRGAN – an SR implementation which focuses on perceptual rather than objective image quality assessment [56]. The advent of the Generative Adversarial Network (GAN) allowed the development of learned perceptual-based loss for image SR. Minimizing pixel-wise reconstruction error is not always the goal of an SR algorithm. Sometimes it is only of interest to make the image "look better" (according to a human observer) relative to alternatives. For this subset of SR tasks perceptual-based losses are used, which focus on enforcing natural image features [56, 101]. In their quest to minimize pixel-wise error, objective-based losses tend to make fine detail overly smooth, washing out high frequency detail. By ignoring or regularizing/weakening the pixel-wise error constraint, perceptual-based losses can generate realistic *looking* textures with fine synthetic details [56]. SRGAN was built for this purpose. Whether a given generative SR convnet will produce an objective-based output or a perceptual-based output depends on the selection of loss function, a choice which is entirely task-dependent. Although SRResNet was initially used as the generator alongside an adversarial perceptual loss, SRResNet can be paired with an objective-based loss function and can be used as a stand-alone generative SR convnet with better mean-squared-error reconstruction performance than SRCNN [56]. For super-resolution on medical images (where synthetic data can potentially change an assessment with life-or-death consequence) the generation of spurious image data should be avoided. The asymmetric super-resolution module I develop in Chapter 4 can be used for either objective- or perceptual-based losses, but my implementation uses an objective L1-pixel loss function because I intend to use it on medical images. Most objective-based SR convnets in the literature have been trained with either L2- or L1-pixel loss as the primary loss function [20].

The more recent SR convnets improve reconstruction performance even further. This is largely attributed to their wider (with more learned features per convolutional layer) and deeper (with more convolutional layers, most often in the form of ResBlocks) architectures [20]. Two of the most performant of these networks are EDSR [102] and RCAN [35]. EDSR improves the SRResNet architecture by making the network twice as deep (at 32 ResBlocks or 64 individual convolutional layers) and four times as wide (with 256 feature maps per layer). They also remove the batch normalization layers from the ResBlocks, which they found eased the training of the network [102]. This latter change is replicated in most of the more recent SR convnets [20]. RCAN implements an even deeper network (with 200 ResBlocks or 400 convolutional layers) and adds a channel-attention mechanism. This increased depth is accomplished by implementing “residual-in-residual” connections. Simply, these are skip connections that circumvent parts of the network at various scales, allowing information to flow around large parts of the overall network structure.

Research into generative SR convnets continues today, with themes of dense residual connections [103, 104], improvements to the attention mechanisms [105], and magnification-arbitrary models [106] which can super-resolve at multiple integer scale factors in a single model. One aspect that remains mostly unexplored is the case of asymmetric super-resolution. Each of the works described above solely focuses on training and testing SR convnets for symmetric scaling, where each dimension of the image is scaled by the same scaling factor. Alternatively, *asymmetric* SR seeks to scale the image at a different scaling factor for each image dimension. The Scale-Arbitrary SR module from Wang et al. [21] is the first (and to my knowledge, only) work that implements asymmetric super-resolution. They test their network on a variety of asymmetric scaling situations where each dimension is scaled by a different factor. Notably, in [21] Wang et. al do not test their module on (what I will call) the *purely* asymmetric case where only one image dimension is upscaled while the other is held at unity – that is the subject of Chapter 4 of this thesis.

Chapter 3
Multi-View Ensemble U-Net

The first step in the segmentation framework I develop is semantic segmentation of the column of vertebral bodies. I use semantic segmentation as a steppingstone to the ultimate task of instance segmentation of the individual vertebral bodies. Presently convnet-based segmentation methods dominate the image segmentation task [2]. I implement a modified U-Net [12] to perform this segmentation. While 3D U-Net-like architectures have been used effectively on 3D image data, they have much higher computational complexity and larger memory footprints [18, 19]. Inspired by the performance of 3D convnets on semantic segmentation of vertebrae from CT scans [28], I initially attempted to use a 3D U-Net [18] for the vertebral body segmentation task. I found that the computational requirements of that architecture were too great for my hardware (an Nvidia GTX 1070 GPU with 8GB of memory). Therefore, I focused on a pseudo-3D approach like that used by Shigeta et al. [91], consisting of an ensemble of 2D U-Nets, each trained from a different anatomical “view” or projection of the image volume. In contrast to [91], the model I implement is trained on three object classes: the vertebral body, pelvis, and sternum bone structures. As mentioned in Chapter 2, these bone structures contain high levels of stem cell proliferation post- hematopoietic stem cell transplant (HSCT).

3.1 Methods

3.1.1 Dataset

Being at their core supervised learning algorithms, the training dataset is a critical component of any convnet. The medical image data used in my research is the same used by Williams et al. in their full-body FLT-PET HSCT imaging pilot study [6]. The dataset used in these works consists of joint CT/FLT-PET scans of 23 hematopoietic stem cell transplant patients. In this thesis, I use image volume data from 22 of the 23 original patients. The dual-modality imaging procedure allows localization of bone structure from CT volumes and subsequent measurement of cell proliferation in the associated marrow cavities via the FLT-PET volumes [10, 11]. All patients underwent myeloablation to obliterate their existing bone marrow (including hematological malignancies) prior to hemopoietic stem cell transplantation [6]. The patients were imaged on multiple days during their treatment: on the day before stem cell transplantation (when the patient's haemopoietic system has been ablated), between 5 and 9 days after transplantation, and on the 28th day after transplantation. These intervals allowed the authors of [6] to track the growth patterns of hematopoietic stem cells (HSCs) in the marrow cavities as they progressed towards engraftment.

Enabling the use of a convnet for this segmentation task are ground-truth segmentation masks. These were created manually by a non-physician engineer for three bone structures: vertebral body, bone, and pelvis [17]. The annotations were made from the axial scans only. While my work mostly focuses on the vertebral body segmentation task, the availability of the pelvis and sternum masks allows me to showcase of the flexibility of the convnet-based approach to medical image segmentation. These ground-truth masks were provided for only a subset of the 22 patients, as shown in Table 2. Most of the available ground-truth annotations are of the vertebral body class, of which 35 volumes have been annotated from 19 of the patients. The availability of sternum class and pelvis class ground-truth data is more limited. Notably, the volumes which have labelled pelvis masks also have labelled masks for the other bone structures. There was a labelling inconsistency in the ground-truth data where some of the vertebral body class masks contained the coccyx bone structure, but most did not. I fixed this inconsistency by removing the coccyx structure from all ground-truth vertebral body masks in which it was present.

The dataset is particularly challenging for the vertebral body instance segmentation task because the image volumes are under-sampled along the axial direction, making the boundaries between the individual vertebrae difficult or impossible to resolve in the CT modality without prior information. The CT scans were acquired axially with a pixel-resolution of 512×512 pixels per axial scan. The oversized step between axial scans creates anisotropic voxels where each voxel spans approximately three times as far in the axial direction. A glimpse of the dataset is provided in Figure 4, where the under-sampled nature of the volumes is made clear by the sagittal and coronal views; the structures appear to be "squished" by a factor of three. In this example (which is representative of the entire dataset) the separation between the vertebral bodies is not consistently detectable by humans. This evidenced in Figure 4 by the mostly-contiguous ground-truth mask which contains only one visible vertebral boundary found by the annotator. This aspect of the dataset is further exemplified in Figure 5, where the boundaries between

Table 2. Available Ground-Truth Volumes for the CT/FLT-PET Dataset

	Vert. Body	Sternum	Pelvis	All 3 Classes
# patients	19	7	6	6
# volumes	**35**	**21**	**16**	**16**

the lumbar vertebrae are visible, but there is not enough resolution to accurately discern the boundaries between many of the thoracic and cervical vertebrae.

The FLT-PET data was captured axially at a pixel-resolution of 144 × 144. Notably, the FLT-PET data has a higher sample rate than the CT data along the axial dimension. The CT volume depicted in Figure 4 has dimensions of 512 × 512 × 188 pixels, while its associated FLT-PET volume has dimensions of 144 × 144 × 238 pixels. Since the CT and FLT-PET volumes span approximately the same length in the axial direction, the FLT-PET volumes have slightly higher resolution in the axial direction. However, the FLT-PET volumes have much lower resolution on the axial plane than their associated CT volumes. The additional axial-direction resolution of the FLT-PET scans is exploited in Chapter 5 to assist in the instance segmentation of individual vertebral bodies.

An impactful characteristic of the dataset is that it is class imbalanced. A class imbalanced dataset is one where the class examples are not equivalent in number to each other (or the background class, in the case of image segmentation). This is a common challenge in learning-based semantic segmentation, where often the objects of interest can be small relative to the background class [107, 108]. In this dataset, the spine mask makes up approximately 0.15% of all voxels, the pelvis mask 0.25%, and the sternum mask only 0.019%. This means that around 99.5% of voxels in a given CT scan will make up the background class. Aside from the raw volume imbalance of class voxels vs. background voxels, the dataset is imbalanced in another way when used in a 2D or pseudo-2D fashion: often the 2D slices of the 3D image data will not contain any of the object classes at all. This is particularly true in the sagittal and coronal views of the spine. The spine is obviously elongated in the axial direction, and a randomly chosen axial scan has a relatively high likelihood of containing spine-class voxels. However, in the sagittal and coronal views only a fraction of 2D slices contain any spine-class voxels at all. These imbalances motivate changes to the loss function, discussed in Section 3.1.3.

3.1.2 Model Design

I use a lightly-modified version of the original U-Net [12] for the constituent convnet models in the ensemble. The only architectural difference between this implementation

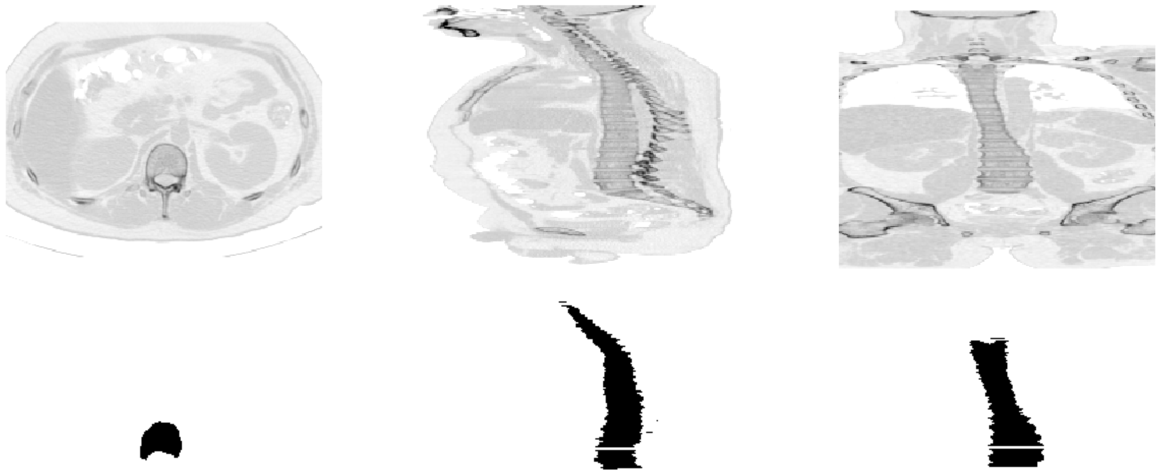


Figure 4. An exemplar axial slice and its associated ground-truth vertebral body mask (left), an under-sampled sagittal slice (middle), and an under-sampled coronal slice (right).

and the original U-Net (shown in Figure 3) is the incorporation of a batch normalization layer [55] after each 3 × 3 convolution. The addition of batch normalization layers to the U-Net architecture is suggested in [18] for faster convergence during training. The PyTorch model was adapted from [109]. Three models are trained independently from each of the axial, sagittal, and coronal views on the three bone structure object classes. The resulting models are then used in an ensemble configuration to provide a segmentation for 3D input volumes. In this configuration, an input volume is sampled sequentially in 2D slices along the appropriate plane (axial, sagittal, or coronal depending on the model). The slices are fed into the U-Net model and the prediction outputs are sequentially recombined to create a prediction volume. In this way, a single input volume is used to generate three prediction volumes, one for each of the axial, sagittal, and coronal models. The prediction volumes are image volumes where the intensity of each voxel (ranging from zero to one) represents a confidence grade that the pixel belongs to a certain class (in this case, a particular bone structure). In cases where a model has multiple object classes the prediction image has multiple channels, each representing a unique object class. In my experiments I use the vertebral body, pelvis, and sternum bone structures as the object classes. There are a few ways to combine the models to create the final inference volume. Shigeta et. al [91] use only one object class (the vertebrae) and combine their three constituent models by voxel-wise averaging of three prediction volumes. Given a multi-class input image volume I and the predictive U-Net models U_A (trained on axial image data), U_S (trained on sagittal image data), and U_C

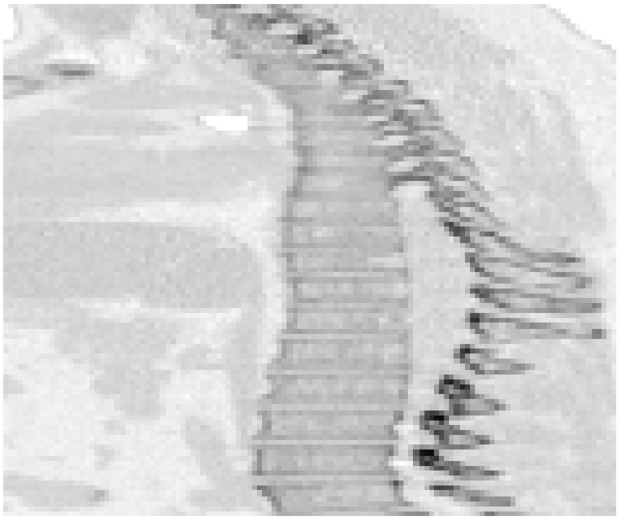


Figure 5. Close-up of an under-sampled sagittal view of the spine.

(trained on coronal image data), the ensemble-by-averaging model E_{avg} can be described by

$$E_{avg}(I,c,p) = T\left(\frac{U_A(I,c)+U_S(I,c)+U_C(I,c)}{3},p\right) \ , \tag{12}$$

where T is a thresholding function with threshold p. The result E_{avg} is a volume image mask containing the 3D semantic segmentation of the object class c. In this chapter I test the above ensemble-by-averaging strategy employed by [91], but I also test two others. First, an ensemble-by-voting strategy where each raw prediction volume $U_A(I,c)$, $U_S(I,c)$, $U_C(I,c)$, is thresholded individually by p to create its own prediction mask $M_X(I,c,p) = T(U_X(I,c),p)$ which casts a vote as to whether a given voxel belongs to a given object class. This strategy is described voxel-wise by

$$E_{vote(i,j,k)}(c,n) = V_{(i,j,k)}(M_{A(i,j,k)}(I,c,p) + M_{S(i,j,k)}(I,c,p) + M_{C(i,j,k)}(I,c,p),n) \ , \tag{13}$$

where V is a "voting" transformation applied voxel-wise to the sum of the thresholded volume masks of class c, and a voxel at coordinates (i,j,k) is assigned a positive class label in the prediction volume only if it has received enough "votes" n from the constituent predicted masks in the ensemble:

$$V_{(i,j,k)}\left(\widehat{M}_{(i,j,k)},n\right) = \begin{cases} 1 \ , & \widehat{M}_{ijk} \geq n, \\ 0 \ , & \text{otherwise.} \end{cases} \tag{14}$$

In my experiments I use $n = 2$ for the ensemble-by-voting method, so that a voxel is considered an object voxel only if a majority of the three constituent U-Nets have predicted so. The last ensemble method I test is an "additive" method. Using this method, if any of the models predict an object pixel it is included in the final prediction mask. It is essentially just ensemble-by-voting but with $n = 1$.

3.1.3 Loss Function

Cross-entropy (CE, also "negative log-likelihood") is an obvious and popular choice for multiclass semantic segmentation [110], but other loss functions have been recently suggested specifically for use in class-imbalanced datasets, such as focal loss, which (in the case of pixel-wise semantic segmentation) down-weights well-classified pixels [111]. Other interesting choices that have been effective on imbalanced semantic segmentation datasets are the "soft" Dice loss [112] and Lovász-softmax loss [113], which aim to reflect the Dice and Jaccard metrics (respectively) better than the common surrogate of CE loss, while maintaining differentiability for backpropagation (hence "soft").

Another option to handle class imbalance is to simply use a weighted cross-entropy loss and tune the class weights to under-value the background class while increasing the weights on the disproportionate object classes [114]. This is the approach I use, for two reasons. First, initial testing on a single-class segmentation task (only vertebral body class) showed similar validation set convergence for CE and Focal Loss. Second, due to the relative ease of implementation for the multi-class case. The weighted cross entropy loss for multiple object classes is given by [40, 114]

$$\mathrm{WCE} = -\sum_{i=1}^{c} w_i t_i \log p_i \, , \tag{15}$$

where c represents the number of object classes (including the "background" class in the case of image segmentation), w_i is a weight assigned to each object class, t_i is the ground-truth class indicator with a value of 0 or 1, and p_i is the model's prediction confidence level on the object class ranging from zero to one. For an image, this loss is calculated per-pixel and aggregated. In training a classification neural network this loss is often used on a minibatch of prediction/target training pairs, where backpropagation is performed from the mean WCE of the minibatch [32].

3.1.4 Training Scheme

The dataset is limited and contains multiple CT image volumes for each patient, so it also has some redundancy. In such situations, care must be taken to ensure that no patient appears in any two of the training, validation, or test set. There are only six patients with all three classes (vertebral body, pelvis, and sternum) labelled. I use 14 CT image volumes from five patients for the training set. Two image volumes from one patient are used for multiclass validation. Quantitative testing is performed on only the vertebral body class, which allows for a much larger than typical test set of 13 volumes from 13 patients. The choice to use only five patients for training and 13 for testing may seem odd but is motivated by the desire to show a flexible multi-class semantic segmentation for bone structures. Since there are only six patients with all three ground-truth mask volumes, I decide to use all but one of those for training, use the single remaining patient's image volumes for validation, and test the model on only the vertebral body segmentation, for which there are many more ground-truth masks from the other patients. I still validate the model on the pelvis and sternum segmentations and show qualitative results for those object classes to showcase the flexibility of the U-Net for semantic segmentation.

The volumes are sliced into 2D slices prior to training. Depending on the model, they are sliced along the axial, sagittal, or coronal plane. Geometric data augmentation is used to increase the diversity of the training dataset [57]. Input images and their target masks are scaled from -20% to 20%, random rotation is applied from -45 to +45 degrees, horizontal and vertical translation is applied up to 20% of the image height and width, and horizontal and vertical flipping are applied. Training images are also cropped to a size of 320×320 pixels, with padding-by-mirroring to account for missing image data that may arise from the geometric augmentation. All augmentations, excepting the final crop, are performed randomly with a uniform distribution.

For training the models I use the Adam optimizer [44] with $momentum = 0.9$ and $betas = (0.9, 0.999)$, and an initial learning rate of 10^{-4}. Final models were trained over 32 epochs where in each epoch they saw every 2D slice of 3D training volumes, for a total of 2579 input training pairs per epoch in the axial-trained model and 7168 input training pairs per epoch in the sagittal- and coronal-trained models. A minibatch size of 6 was used. Validation is performed at regular intervals on the entire validation volumes. The learning rate was reduced by an order of magnitude when the validation failed to reach a new maximum within 8 epochs. When training with CE loss without weighting (all classes have equal weight) the model learns to make positive classifications for object classes slowly. It is likely able to quickly reduce loss by simply predicting background classes prolifically. This effect is shown in Figure 6, where no object classes make positive classifications for at least an epoch, and in the case of the extremely-imbalanced sternum class, for even longer. To mitigate the slow learning I apply class weights w_i to the cross entropy loss (15). I found setting the background class weight to 0.1 and the three object class weights each to 0.3 was effective for encouraging positive classification earlier in the training cycle, to within the first epoch for the vertebral body and pelvis classes. The sternum class, with its extremely-small volume footprint and low number of 2D examples still struggles to achieve positive classification within the first epoch but is greatly improved from the non-weighted CE loss. The effect on the training is shown in Figure 7. Note that the difference in the smoothness of the charts is solely due to a reduction in the frequency of running validation on the model (a purely time-saving measure). These charts depict the axial-view model training, but the same effect is seen in training both the sagittal and coronal models. Notably, I found that intuitively setting class weights to be inversely proportionate to the class sizes in the dataset (for my case, a proportion of voxels) was not effective, generating erratic validation curves and lower validation convergence. Figure 8 charts the validation of all three constituent models (axial, sagittal, and coronal) for the vertebral body class. The coronal-view U-Net model had trouble converging, so it was trained further with a reduced learning rate for an additional 6 epochs. When evaluated individually, the model trained with axial image data achieves the highest validation score on the vertebral body class.

3.2 Results

3.2.1 Quantitative Results

The constituent models were scored individually and in the ensemble configuration. The scores were obtained by using a classification threshold of $p = 0.5$ on the raw prediction volumes and comparing the resultant prediction masks to the ground-truth prediction volume. Of the three constituent U-Net models, the axial-trained model performs the best on the test data with a mean Dice score of 0.922 on the vertebral body class, with the sagittal model close behind at 0.902. The coronal model performed poorly relative to the others with a Dice score of 0.849 on the vertebral body class. The ensemble-by-voting and ensemble-by-averaging methods reduced the Dice score performance to below that of the axial model, both scoring a Dice of approximately 0.914 on the vertebral body class. The additive models (where the predictions from each constituent U-Net are simply added together) saw mixed performance compared with the

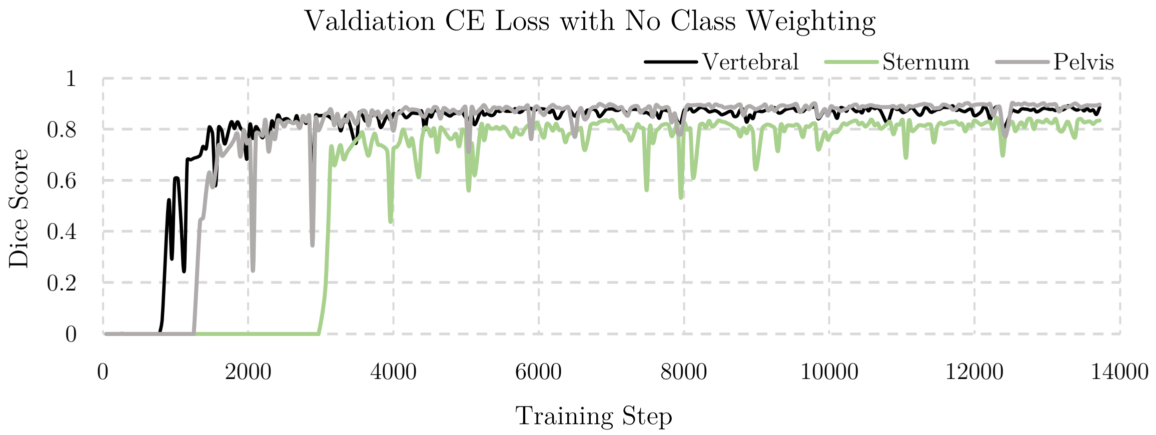


Figure 6. Training (axial view) with cross-entropy loss with equal class weighting. Positive classification is delayed by the class imbalance in the dataset.

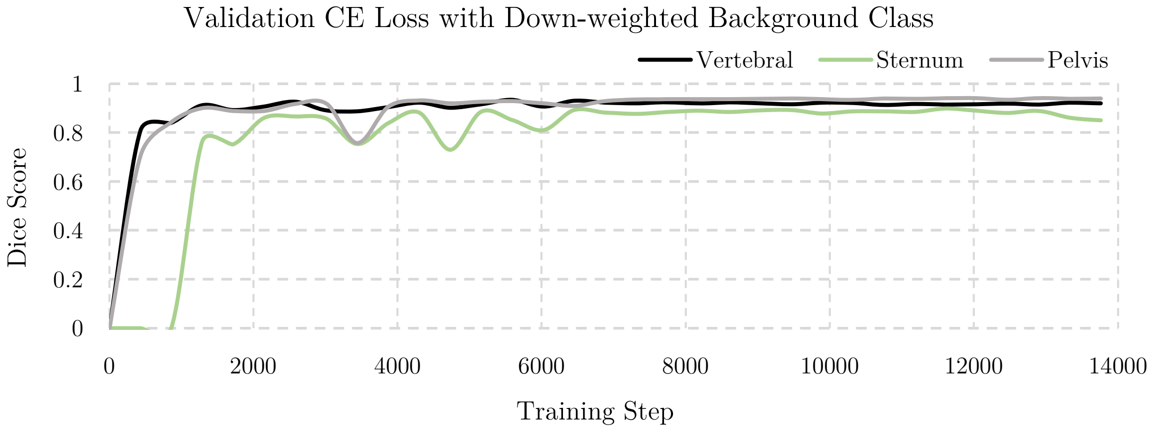


Figure 7. Training (axial view) with weighted cross-entropy loss with down-weighted background class. Positive classification in encouraged by the higher contributions of the object classes to the loss.

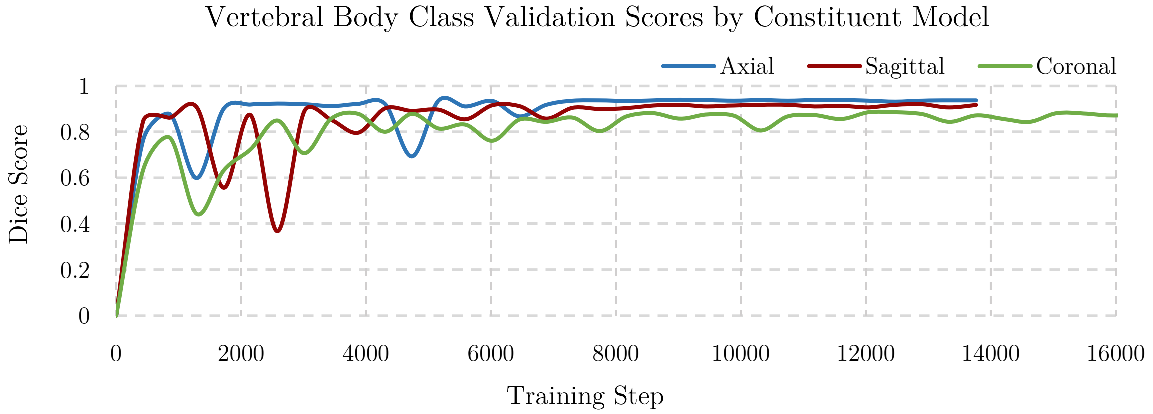


Figure 8. The vertebral-body class validation scores during the training of the 3 constituent (axial, sagittal, and coronal view) U-Net models. The coronal view had issues converging and was further trained with a lower learning rate.

TABLE 3. DICE SCORES FOR THE MULTI-VIEW ENSEMBLE U-NET MODELS ON THE VERTEBRAL BODY OBJECT CLASS

volume	Axial	Sagittal	Coronal	Vote	Avg	Add "3"	Add "2"
p7d3	0.9012	0.8836	0.7620	0.8971	0.8967	**0.9020**	0.9005
p8d3	0.9180	0.8725	0.7576	0.8936	0.8924	0.8940	**0.9214**
p9d3	**0.9342**	0.9094	0.9033	0.9307	0.9303	0.9264	0.9270
p10d3	0.9297	0.9113	0.8899	0.9261	0.9255	0.9269	**0.9328**
p11d3	**0.9380**	0.8925	0.8575	0.9176	0.9207	0.9089	0.9316
p12d3	0.9088	0.9074	0.7890	0.9076	0.9047	0.9190	**0.9194**
p13d3	0.9198	0.9003	0.8696	0.9115	0.9097	0.8999	**0.9216**
p14d3	0.9179	0.9046	0.8579	0.9052	0.9075	0.9221	**0.9242**
p16d3	0.9381	0.9212	0.8910	0.9281	0.9311	0.9323	**0.9390**
p17d3	0.9019	0.8799	0.8692	0.8956	0.8952	**0.9154**	0.9091
p18d3	**0.9200**	0.8962	0.8223	0.9093	0.9082	0.9186	0.9190
p19d3	**0.9350**	0.9171	0.8988	0.9316	0.9323	0.9267	0.9338
p21d3	0.9258	0.9261	0.8674	0.9245	0.9263	0.9221	**0.9387**
mean	0.9222	0.9017	0.8489	0.9137	0.9139	0.9165	**0.9245**

Add "3" is the additive ensemble model with all three views included. Add "2" is the additive ensemble model with only the axial and sagittal views included, where the low-performing coronal model is excluded. ***Bold*** *is best and underlined is second-best.*

axial model. The additive model that incorporates all three prediction volumes scores below the axial model. The additive model using only the axial and sagittal prediction volumes scored the highest mean Dice on the vertebral body class, at 0.9245. The results are shown for all test volumes in Table 3.

It is worth investigating the classification performance of each of the constituent axial, sagittal, and coronal U-Net models independently. To this end, ROC curves were generated by procedurally calculating TPR and FPR across the range of possible classification thresholds from 0 to 1. The TPR and FPR for each model is shown in Figure 9. The ROC curves appear to show a classifier with high sensitivity and specificity, but they may be misleading due to the severely imbalanced segmentation dataset. Since the vast majority (over 99.5%) of voxels belong to the background class, even a tiny increase in the false-positive rate has a large negative impact on the segmentation accuracy as measured by the Dice score. I elect not to report the area-under-curve (AUC) measure due to the class imbalance making it an unrepresentative gauge of segmentation accuracy. The mean distribution of voxel values in the test prediction volumes output from the U-Net models are shown in Figure 10. Each of the axial, sagittal, and coronal

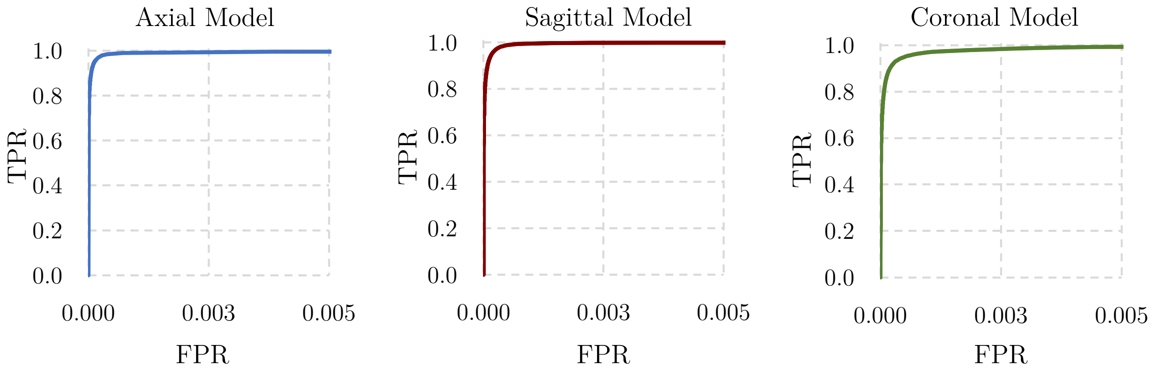


Figure 9. Receiver-operating characteristic (ROC) curves for the three constituent U-Net models. Note that the false-positive axis is limited to the window [0,0.005].

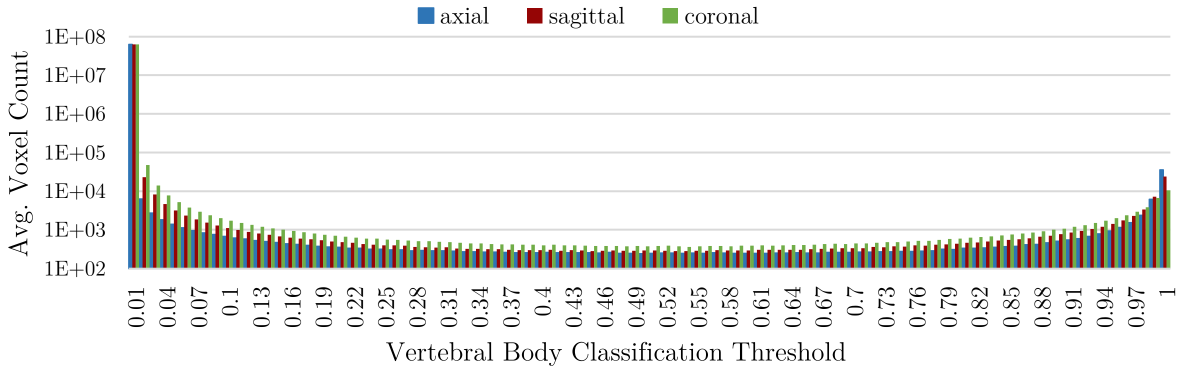


Figure 10. Histograms averaged from the output prediction volumes of U-Net models. The three models show similar distribution. Note that the count axis is logarithmic and begins at a non-zero value.

U-Net models has a similar distribution of prediction confidence levels. Most background class voxels have vertebral body class predictions below 0.01, leading to the high specificity seen in the ROC curves. One could select nearly any threshold, even $p = 0.02$, and still correctly classify most background voxels. It is also notable that the axial-trained model has a higher proportion of class predictions in the "very high confidence" range between 0.99 and 1.0.

An interesting result found in the raw prediction volumes of the test set is that higher Dice scores could be achieved by setting a different classification threshold than the naïve choice of $p = 0.5$. The average performance gains are small for the higher-performing axial and sagittal models, but for the comparatively mediocre coronal-view U-Net model the gains are high, as can be seen in Figure 11. The mean optimal threshold values for the models on the test set are $p = 0.427$ (axial), $p = 0.323$ (sagittal), and $p = 0.295$ (coronal). If these "test set optimal" threshold values were used for classification, the mean test set Dice scores on the constituent U-Net models would be approximately 0.925 (axial), 0.911 (sagittal), and 0.875 (coronal). These performance gains, since they

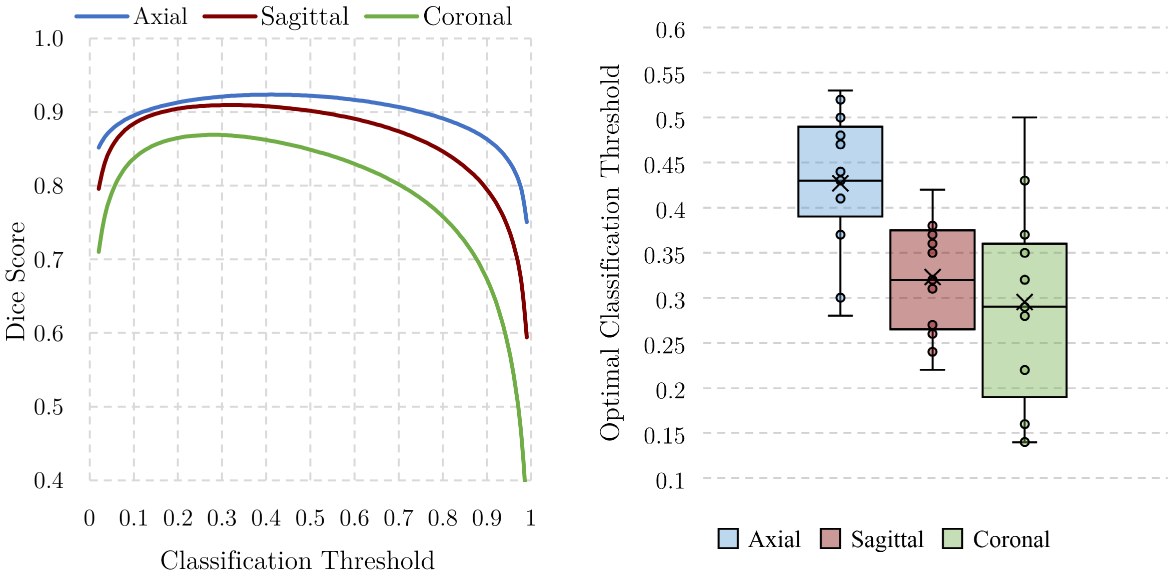


Figure 11. (Left) Mean test set Dice score vs classification threshold level. (Right) Boxplot of optimal classification thresholds for entire test set for each constituent model. "×" indicates the mean value. Whiskers extend to largest/smallest value within 1.5 times the interquartile range.

depend on prior knowledge of the test set, should only be taken as evidence of a trend: lower classification thresholds may lead to improved Dice scores for these U-Net models. The distribution of the optimal thresholds of the test set make it very likely that the true mean optimal classification thresholds lie below $p = 0.5$ on CT data similar to the test set. Assuming a normal distribution with unknown variance, the 99% confidence intervals for the mean optimal classification thresholds are [0.38, 0.47] for the axial model, [0.28, 0.36] for the sagittal model, and [0.23, 0.36] for the coronal model. The performance of the ensemble models at these optimal thresholds remains untested since the result would be based on prior knowledge and would not be representative of performance on unseen data.

3.2.2 Qualitative Results

The predictive capabilities of the constituent axial-, sagittal-, and coronal-view U-Net models on the vertebral body class are visualized below in Figure 12. The axial model is obviously superior in predicting the vertebral body shape with high confidence and consistency, but all three models localize their predictions well. The sagittal and particularly the coronal model have higher variance in the confidence level within the vertebral body. Confidence levels fall sharply outside of the ground-truth mask region, indicative of the high specificity in each of the constituent segmentation models. The noise-like effect seen in some sagittal and coronal predictions in Figure 12 is attributed to these being reconstructions of the axial view for these models. Figure 13 and Figure

14 show similar prediction slices but reconstructed in the sagittal and coronal views. Notably, the sagittal view reveals that the models are able to detect a few of the boundaries between the vertebrae, even when the human annotator was not able to discover them in the ground-truth. This is likely due to human inconsistency in the difficult task of labelling the vertebral bodies. Some ground-truth masks (e.g., Figure 4) contain one or more identified vertebral boundaries in the lower lumbar region, and the model has learned from these examples. The sagittal views of Figure 13 also reveal the relatively poor performance of the coronal model.

A visualization of the 3D vertebral body segmentation volumes from the additive, axial, sagittal, and coronal models are shown in Figure 15 with false positives shown in red and false negatives shown in blue. True positives are grey. The visualization tends to perceptually overstate the presence of classification errors since the false classification surfaces occlude the true positive surface. Still, it is a useful tool for analyzing the qualitive classification performance of the models. At the $p = 0.5$ classification threshold the models are more prone to false negatives than false positives. This visualization also makes it easy to understand the effect of the additive ensemble models on true positives and true negative predictions. False negatives from the constituent models only persist through the additive model if the false negatives appear in *all* constituents. Conversely, false positives will persist if they appear in *any* of the constituent models. This means that the additive model will only be effective if the constituent models tend to generate more false negatives than false positives – which appears to be the case with these models and the average test set CT volume. Lastly, the visualization is also useful for revealing errors in the ground-truth volume. Intermediate analysis of the prediction volumes showed ground-truth labelling inconsistencies between the training and test sets. Specifically, the volumes selected as test set volumes contained the coccyx (colloquially called the tailbone) as part of the vertebral body mask, while the volumes used for training the U-Net models did not include the coccyx in the vertebral body class. This issue was only revealed by indications of the coccyx bone structure appearing as false negatives on the test set, as shown in Figure 16. I fixed the inconsistency by removing the coccyx from the vertebral body class masks of the test set volumes. The quantitative results presented earlier in this chapter are from the corrected test set volumes.

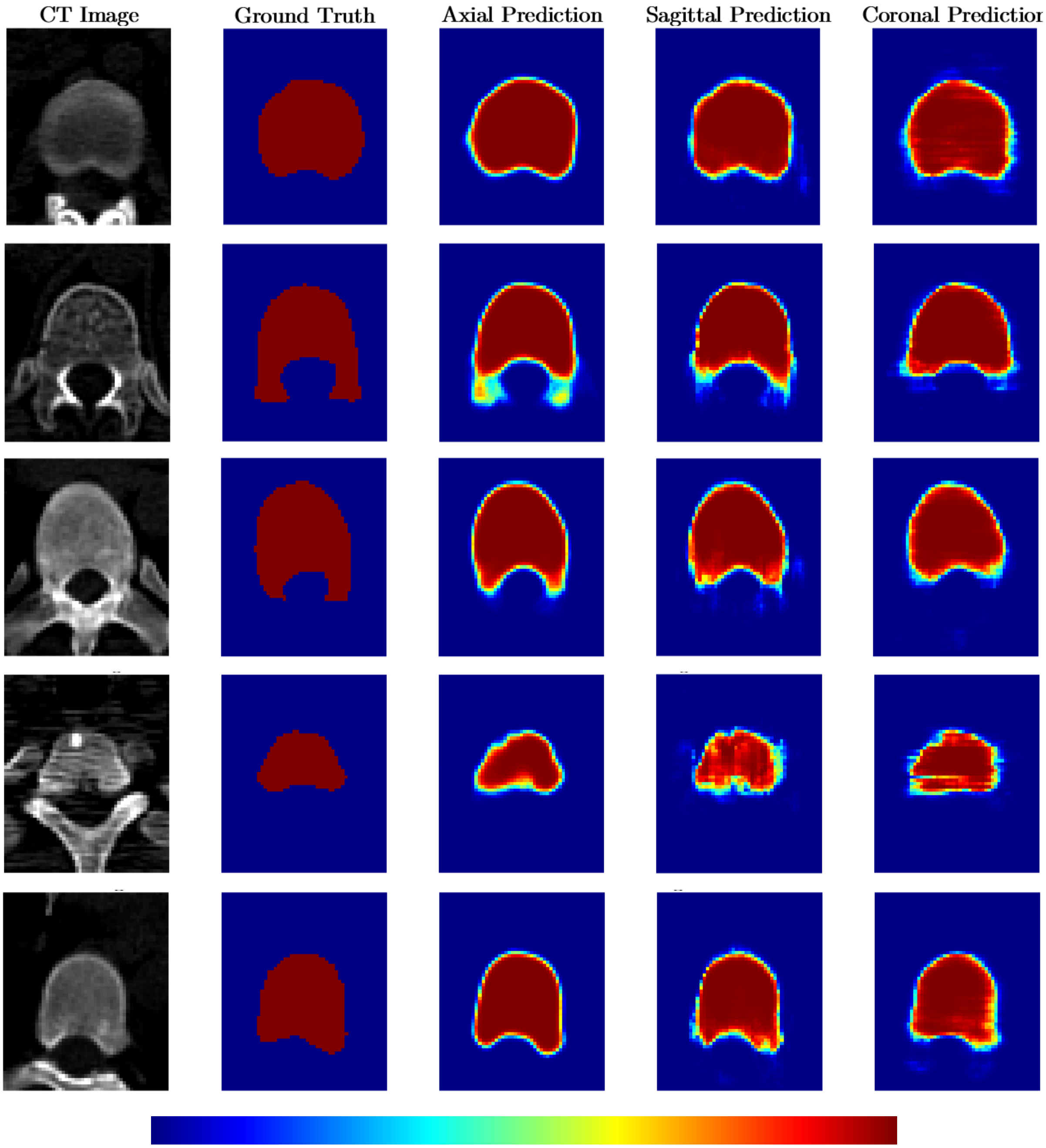


Figure 12. U-Net output predictions of the vertebral body class from the constituent axial, sagittal and coronal U-Net models on a collection of test volume inputs. Confidence scores range from zero (dark blue) to one (dark red).

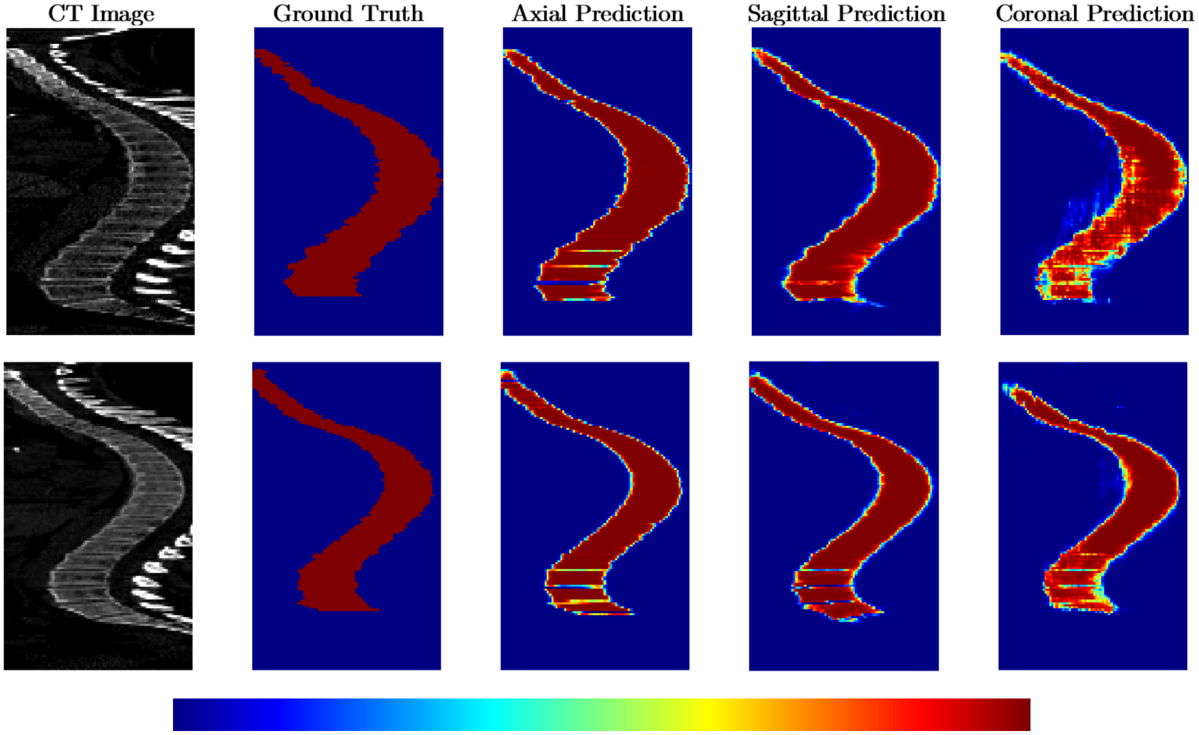


Figure 13. U-Net output-predictions of the vertebral body class (reconstructed in the sagittal view) from the constituent axial, sagittal and coronal U-Net models on a collection of test volume inputs. Confidence scores range from zero (dark blue) to one (dark red).

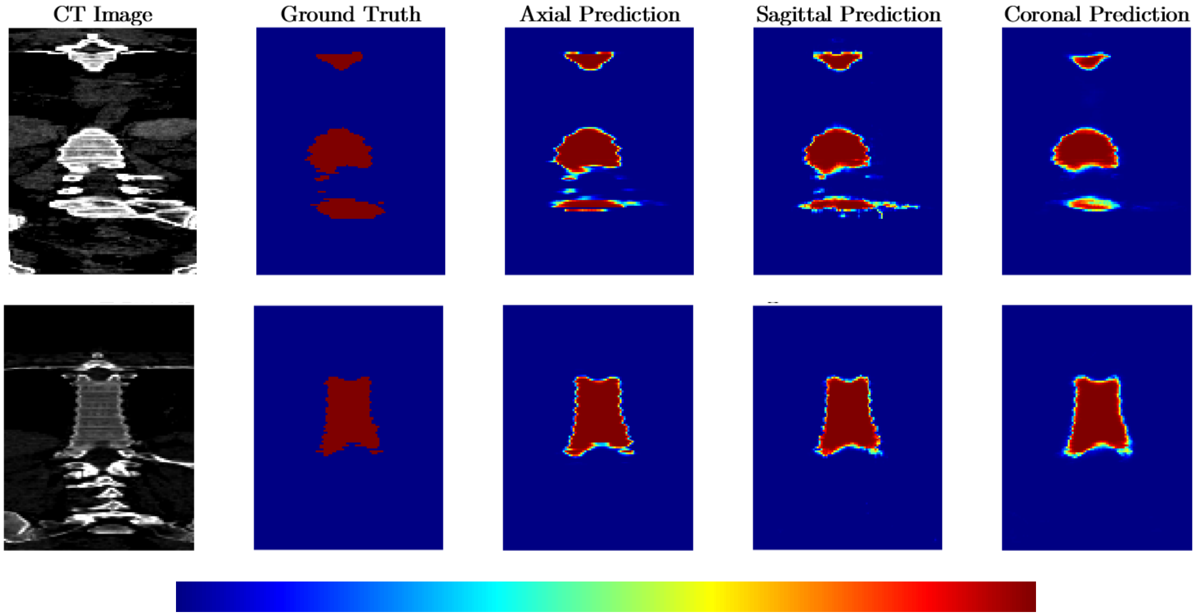


Figure 14. U-Net output-predictions of the vertebral body class (reconstructed in the coronal view) from the constituent axial, sagittal and coronal U-Net models on a collection of test volume inputs. Confidence scores range from zero (dark blue) to one (dark red).

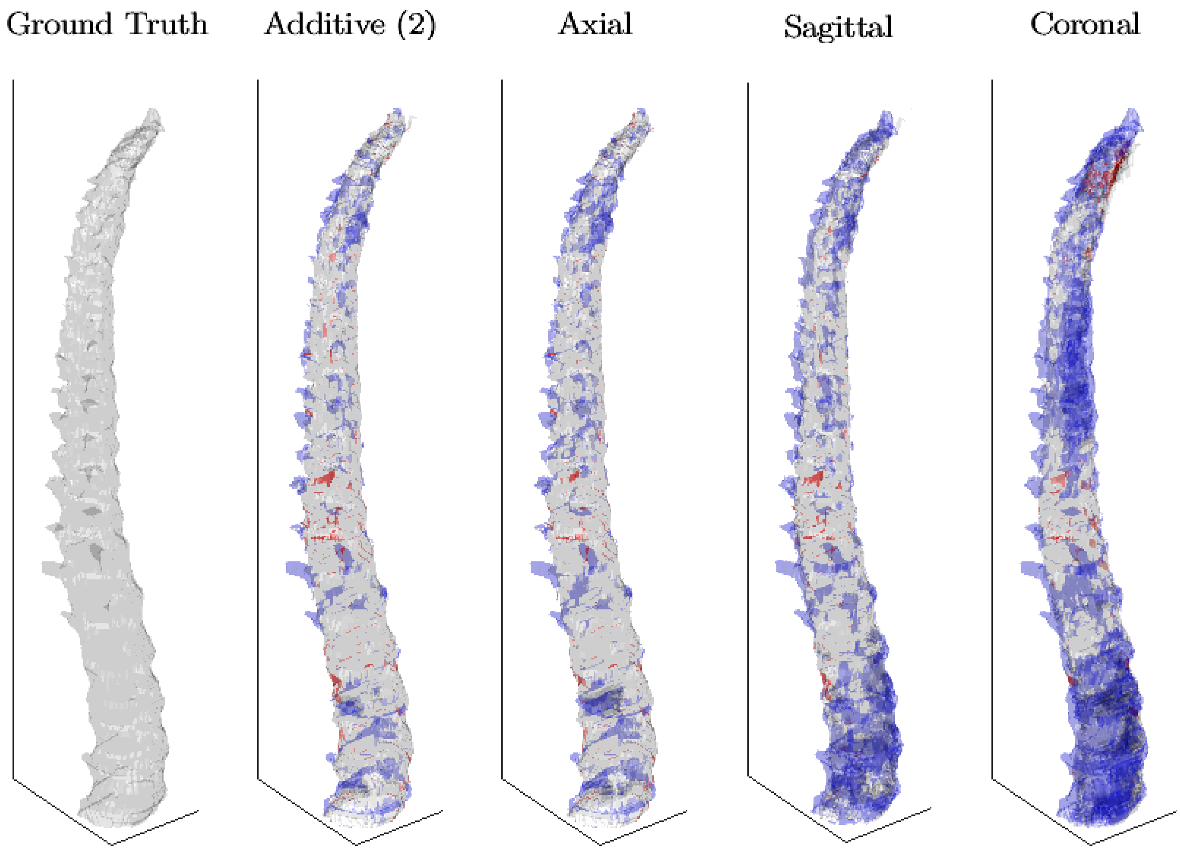


Figure 15. Predicted volume masks at the $p = 0.5$ classification threshold for various models. Additive (2) is the Axial + Sagittal additive model. Grey indicates true positives, red indicates false positives, and blue indicates false negatives.

Finally, I present some qualitative results for the pelvis and sternum classes to highlight the flexibility of the deep learning-based convnet approach to the image segmentation task. Figure 17 shows the pelvis class prediction from the axial U-Net model on the validation volume, and Figure 18 shows the sternum class. Even the singular 2D model does a good job of segmenting the very irregular shape of the pelvis and the small-volume footprint of the sternum. More qualitative segmentation results for the vertebral body, pelvis, and sternum classes are provided in Appendix A for the rest of the test set CT volumes. Some predictions show small groups false positive results far from the bone structures of interest, indicating segmentation could be further improved by morphological filtering such as close/open filters or connected component extraction. Some of these morphological operations are explored in Chapter 5 as a preprocessing step in the vertebral body instance segmentation.

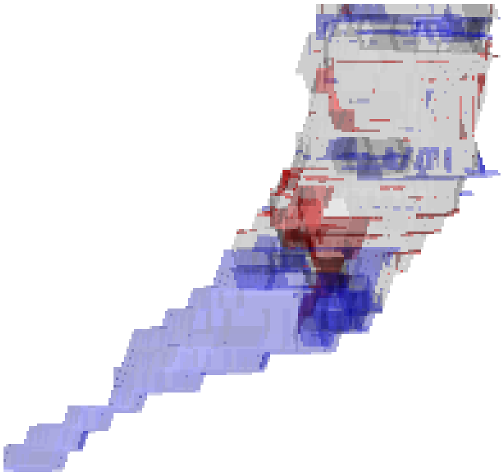


Figure 16. False negatives (blue) and false positives (red) near the coccyx tailbone structure of a test set volume. The persistent false negative pattern revealed an underlying labelling inconsistency between the test set and the training set (which has since been fixed).

3.3 Discussion

In this chapter I presented a multi-view ensemble U-Net model for segmenting vertebral body, pelvis, and sternum bone structures from CT volume data. Due to the limited ground-truth data for the pelvis and sternum classes, quantitative results were reported for only the vertebral body class. Of the constituent axial-, sagittal-, and coronal-view models the axial-view model performs the best with a mean Dice score of 0.922 on the test set for the vertebral body class. It may be argued that the vertebral body segmentation task, when limited to 2D, is easier in the axial plane than the others. The result matches intuitive reasoning. The shape of the 2D masks is much more regular in the axial plane compared to the sagittal or coronal planes. Conversely, due to the typical shape of the spine, slicing 2D images along the coronal plane creates discontinuous and sporadic vertebral body images and class masks, the features of which are naturally more difficult to learn.

On the vertebral body class, the proposed ensemble-by-voting and ensemble-by-averaging models performed worse than even the singular axial-view model. The "additive" ensemble method which preserves the true positives of the constituent models and filters the true negatives did perform slightly better on the test set when used in the "axial-view model plus sagittal-view model" configuration. However, the small performance increase of +0.0023 Dice score on the test set hardly seems worth the extra effort of generating additional U-Net models. I also show that with a reasonable degree of certainty, the mean optimal classification threshold is likely below the naïve choice of

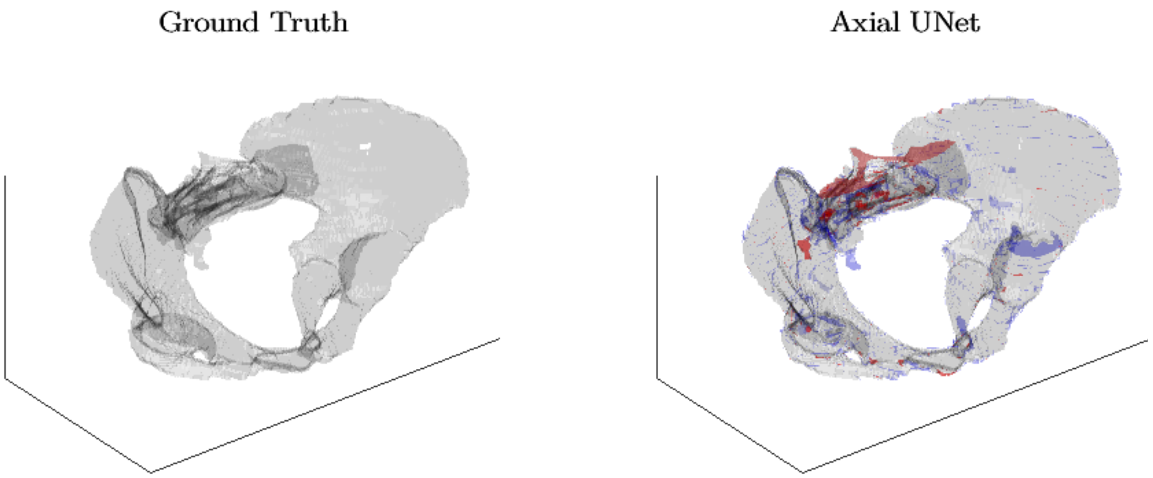


Figure 17. Ground-truth pelvis class mask and a prediction of the pelvis class from the axial-view U-Net model. True positives are grey, false negatives are blue, and false positives are red.

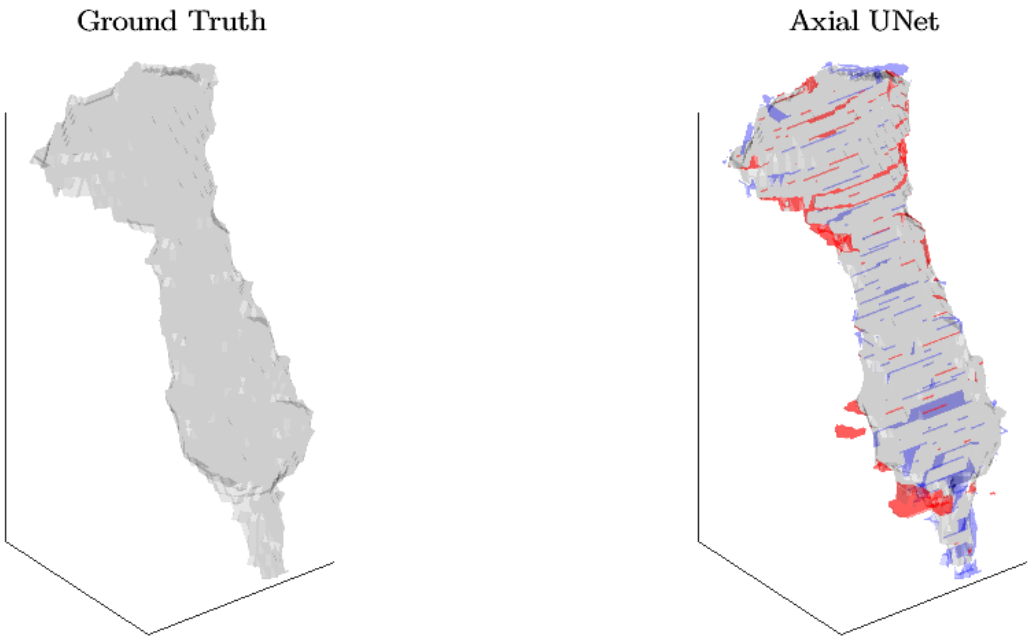


Figure 18. Ground-truth pelvis class mask and a prediction of the sternum class from the axial-view U-Net model. True positives are grey, false negatives are blue, and false positives are red.

$p = 0.5$ for CT image volumes similar to those of the test set. Qualitative results in the form of thresholded model output predictions are provided in Appendix A. These show the effectiveness and flexibility of even the singular axial-view 2D U-Net for the 3D bone segmentation task.

Chapter 4
Asymmetric Super-Resolution

In Chapter 3 it is shown that the under-sampled nature of the HSCT patient dataset makes it difficult to identify the boundaries between individual vertebrae to perform instance segmentation of the vertebral bodies, even for human annotators. In *this* chapter I investigate the use of super-resolution as a preprocessing step for such under-sampled medical image data, with the hope that it may be able to accurately reconstruct anatomical "texture" to the point that these vertebral boundaries may be revealed.

Super-resolution as implemented in the research literature nearly always uses symmetric scaling where both dimensions of the low-resolution image are scaled equally [20]. This type of scaling assumes that the pixels in a low-resolution digital image represent square areas in the real image plane. This assumption is very reasonable in nearly all SR use cases. However, there are cases where *asymmetric* scaling may be useful. Specifically, in 3D medical imaging a single dimension is sometimes under-sampled relative to the others to limit the overall radiation dose experienced by a patient [4]. In these cases the resultant voxels are anisotropic in shape – they are discrete representations of non-cubic volumes in the real image space. When this under-sampled volume is sliced and viewed in 2D, two of the three anatomical image planes will contain asymmetric pixel data (the images in these planes appear to be "squished"). For example, in a case where the axial dimension is under-sampled, the sagittal and coronal views will have anisotropic pixels. To view the under-sampled sagittal and coronal image data in a realistically proportionate way it needs to be re-scaled. Scaling can be done with no prior information by naïve interpolation-based methods, but super-resolution algorithms outperform interpolation-based scaling and will yield a more accurate reconstruction [20].

Motivated by the medical imaging use case, I propose an asymmetric SR upsampling module that allows many of the modern state-of-the-art convnets to be used in a novel way – with purely asymmetric scaling. I define *purely* asymmetric scaling to be scaling of 2D images in only a single dimension, in contrast to the generally asymmetric method of [21]. The module I develop could be generalized to 3D for networks operating with 3D image data and 3D convolutions. I show the effectiveness of the asymmetric upsampling module by implementing it alongside a modified SRResNet [56]. The architectural similarities between many of the modern SR convnets (discussed in the following section) allow this module to “drop into” top-performing architectures such as SRGAN [56], EDSR [102], and RCAN [35] – to name a few.

4.1 Methods

4.1.1 Purely Asymmetric Super-Resolution Upsampling Module

In many modern SR convnets purely asymmetric scaling can be implemented by replacing the upsampling module. Most networks have put this upsampling module at the end of the convolutional ResBlocks, allowing features to be extracted in the computationally less expensive low-resolution space. This design trend is shown in the SR convnet survey from Wang et al. [20]. They also show that most of these networks use the sub-pixel upsampling module first implemented in [100]. This upsampling method has been preferred because it is much more efficient than the alternative of transposed convolutional layers. Presently, the PyTorch [53] and TensorFlow [54] implementations of the sub-pixel layer (*nn.PixelShuffle* in PyTorch) do not allow for asymmetric scaling factors. It is possible to implement the sub-pixel layer with purely asymmetric integer scaling, but investigation of the source code reveals that the functionality of the module lies embedded in C++ code. For this proof-of-concept, at this time I am more interested in a drop-in solution based purely on Python code. I stay in the PyTorch environment by using transposed convolution as the upsampling method. This choice comes at the cost of computational complexity, but with no SR reconstruction performance impact, as the authors of the original sub-pixel upsampling paper show the functional equivalence of the two methods in their addendum [115]. The transposed convolutional layer (*nn.ConvTranspose2d* in PyTorch) implicitly allows for purely asymmetric scaling factors by allowing asymmetric stride and kernels [53]. I define a *purely* asymmetric scaling factor s_p as being the integer scaling factor applied to only one of the image dimensions. I emphasize *purely* asymmetric because in [21] they define the asymmetric scaling factor to be the more general case of a different scaling applied to each image dimension, one of which may or may not be unity (the authors do not test the purely asymmetric case in that paper).

The purely asymmetric upsampling module I propose is a very simple design consisting of two layers (depicted in Figure 19):

- **Layer 1:** An initial convolutional layer used for expanding the number of feature channels prior to the mapping from low- to high-resolution space. This design choice is inspired by the initial convolution in the sub-pixel upsampling module [100]. A 3×3 kernel is used with unit stride. For this layer, the number of output feature channels is set to equal the number of input feature channels multiplied by the purely asymmetric scaling factor s_p. In this way the upsampling module becomes wider for larger scaling factors. The feature channels output from this layer are used to map into the SR space by the transposed convolution *Layer 2*.
- **Layer 2:** A transposed convolutional layer with asymmetric stride and asymmetric kernels. The asymmetric stride is set to $[s_p,\ 1]$ and the asymmetric kernels are sized $[s_p, 3]$. To avoid padding the scaled dimension, a patch size with dimensions that are divisible by many integer factors (e.g., 96×96 pixels) may be desirable. Another option is to have the patch size depend on the scaling factor s_p. However, the unscaled dimension will require padding of 1 on each side due to combination of stride and kernel length in that dimension. The number of input feature channels is equal to the number of output feature channels from *Layer 1*. The output of this layer will be the final SR image, so the number of output feature channels will be equal to the number of color channels in the input image (i.e., three for RGB images, one for grayscale).

4.1.2 Model Selection

The module described above can be dropped into many different modern convnet architectures, so long as they use the "post-feature extraction" upsampling method (where the learned features are mapped from low- to high-resolution by a convolutional upsampling layer at the end of the network, after feature extraction [20]) such as the highly performant EDSR [102] and RCAN [35]. EDSR is a large network with over 43 million learnable parameters and takes almost a week to train on my Nvidia GTX 1070 GPU. For testing the purely asymmetric upsampling module I have decided to use a modified SRResNet (similar to the "baseline" model in [102]) with:

- 16 standard ResBlocks with no batch normalization layers.
- 64 feature channels in each convolutional layer.
- 96×96 high-resolution patch size for training.
- A purely asymmetric upsampling module as described in Section 4.1.1.

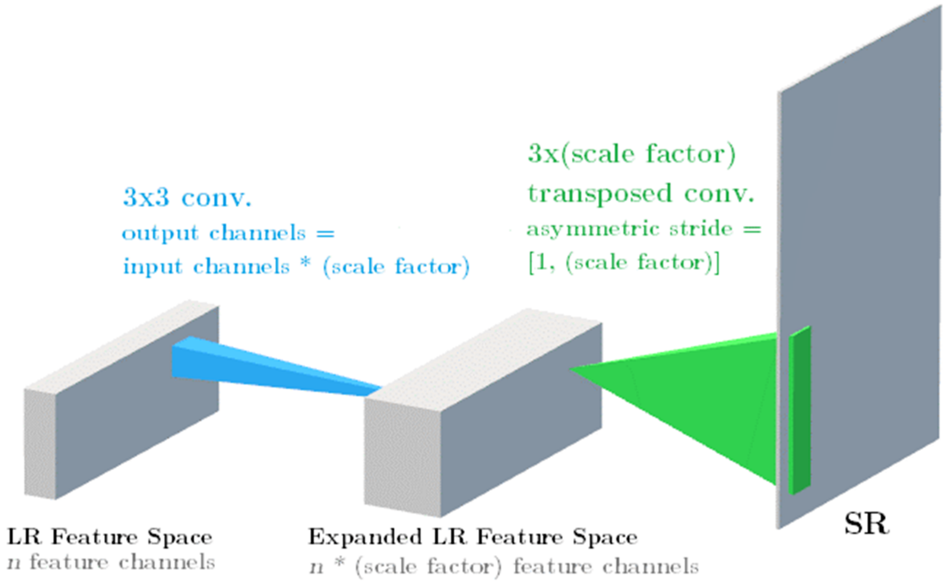


Figure 19. The proposed "drop-in" asymmetric up-sampling module for use in existing SR convnets such as SRResNet, EDSR, or RCAN that use a long chain of residual blocks to learn representations in the low-resolution (LR) feature space. The module is to be placed at the end of the network in place of the symmetric upscaling module.

This model can be trained relatively quickly on my hardware. It is meant to serve as a proof-of-concept for the purely asymmetric SR task, rather than a new or competitive state-of-the-art implementation. My implementation for the modified SRResNet model described above is coded in the PyTorch [53] deep learning framework. I borrow code from the official EDSR for PyTorch Github repository [102] and have made modifications to implement the asymmetric upsampling module for training and inference. This modified implementation can run both the EDSR and the modified SRResNet (and many other SRResNet-based variations) simply by adjusting hyperparameters passed as arguments to the training function. The user can also toggle between purely asymmetric or symmetric scaling modes. I also test the model with the asymmetric upsampling module replaced by the original symmetric sub-pixel upsampling module; this provides a baseline to validate the proposed model against those that appear in the literature (e.g., the "baseline" model used in [102]).

4.1.3 Datasets

Many of the modern SR convnets have been trained on the DIV2K natural image dataset [116]. The dataset contains 800 images for training, 100 images for validation, and 100 images withheld by the authors for competition testing. Since I do not have access to the test set ground-truth I elect to instead split the validation images into two: I use DIV2K images 801-850 for the validation and withhold images 851-900 for testing.

The DIV2K dataset does not contain asymmetric low-resolution images to use for network training, so I generate them from the high-resolution images by bicubically down-sampling the DIV2K dataset by 2x*, 3x*, and 4x* in only the vertical dimension (throughout this chapter, I use the * notation to indicate asymmetric resampling operations). I also create a version of the DIV2K dataset in grayscale for training image SR on a single color channel. The reasoning is that most medical images are single channel, including the under-sampled CT scans from the HSCT patient dataset that I am most interested in applying this network to.

In addition to the DIV2K benchmark, I have created a custom training and test dataset using the publicly available VerSe 2019 CT image volume data [29]. The motivation is to compare the reconstruction performance of an SR model trained on the natural images of DIV2K to that of a model trained exclusively on medical images (when the network is used for re-scaling medical images). I compare the performance of the DIV2K-trained model and the VerSe-trained model quantitatively on a holdout VerSe test set to see if training an SR model on a task-specific dataset confers any reconstruction performance benefit. The highest-resolution CT volumes from the VerSe 2019 dataset were selected to create the 2D VerSe training data. The high-resolution volumes were sliced along the sagittal plane to generate the high-resolution images. The resulting high-resolution sagittal slices were then asymmetrically downsampled in the same way as the asymmetric DIV2K dataset. The asymmetric VerSe training dataset I create contains 800 training images of sagittal views of the spine, 50 validation images of sagittal views of the spine, and a holdout test set contains 10 images of the same. The reason for the relatively small test set is to maximize the training data. To avoid bias, if an image was taken from a volume for the test set, no other images from that volume were used for the training set. Further, each of the 10 images from the test set comes from a unique CT volume, so the 10 images represent 10 entire CT volumes that cannot be used for training.

4.1.4 Performance Metric for Super-Resolution Reconstruction

I measure the reconstruction performance by pixel-wise PSNR, one of the most popular metrics for objectively grading an image reconstruction [20]. This PSNR is based on the mean-squared-pixel-error (MSE) in dB, given by [20]

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[HR(i,j) - SR(i,j)]^2 \ , \tag{16}$$

$$PSNR = 10\log_{10}\left(\frac{I_{max}^2}{MSE}\right) \ , \tag{17}$$

where n and m are the image dimensions (in pixels), HR is the high-resolution ground-truth image, SR is the inferred super-resolved reconstruction, and I_{max}^2 is the maximum possible intensity value allowed by the bit-depth of the images.

4.1.5 Experiments

In all, 17 modified SRResNet models (henceforth just *models*) were trained at different scaling factors, both symmetric and purely asymmetric. The various trained models are used to benchmark on the DIV2K and VerSe training sets. For clarity, Table 4 shows a list of the experiments with their various training sets, with the asymmetric upsampling modules enabled and disabled, and at their various scale factors. PyTorch [53] is used as the deep learning framework. All models were trained from randomly sampled pairs of low- and high-resolution patches from the datasets. High-resolution patches of size 96 × 96 and the corresponding low-resolution patch were used, with the size of the low-resolution patch dependent on the scaling factor and upsampling module. The low-resolution patch is asymmetric for the models trained with the purely asymmetric upsampling module. Basic flip augmentations are performed randomly with a uniform distribution on the training patch pairs, with only the symmetric patches getting 90° rotations (random 90° rotations on the asymmetric low-resolution patches yields size mismatches in the training mini-batch). A mini-batch of 16 patches is used for 300 epochs, where the model sees 16,000 random patch pairs per epoch. The models trained at 4x and 4x* (where * denotes asymmetric) scaling factors used pre-trained 2x and 2x* models to ease training, a method borrowed from [102]. The Adam optimizer [44] is used for training all models, with $momentum = 0.9$ and $betas = (0.9, 0.999)$. L1-pixel is used as the loss function, given by [20]

$$L1(SR, HR) = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|SR_{i,j} - HR_{i,j}\right| \tag{18}$$

for an $m \times n$ ground-truth image HR and predicted image SR, each with spatial coordinates (i, j). An initial learning rate of 10^{-4} was used in all models except the finetuning experiments, which used an initial learning rate of 10^{-5}. This adjustment was made on the finetuning experiments to mitigate overfitting at the 10^{-4} learning rate, as shown by the validation scores in Figure 20. Validation scores for the rest of the models appear in Figure 21, and all show good convergence. A scheduled learning rate

TABLE 4. GENERATIVE SR MODELS TRAINED FOR EXPERIMENT

#	**Training Set**	**Asymm?**	**SF**	**Pre-trained**	**LR**
1	DIV2K	No	2x	-	10^{-4}
2	DIV2K	No	3x	-	10^{-4}
3	DIV2K	No	4x	DIV2K_2x	10^{-4}
4	DIV2K_GRAY	No	2x	-	10^{-4}
5	DIV2K_GRAY	No	3x	-	10^{-4}
6	DIV2K_GRAY	No	4x	DIV2K_GRAY_2x	10^{-4}
7	DIV2K_GRAY	**Yes**	2x*	-	10^{-4}
8	DIV2K_GRAY	**Yes**	3x*	-	10^{-4}
9	DIV2K_GRAY	**Yes**	4x*	DIV2K_GRAY_2x*	10^{-4}
10	VerSe	No	2x	-	10^{-4}
11	VerSe	No	3x	-	10^{-4}
12	VerSe	No	4x	VerSe_2x	10^{-4}
13	VerSe	**Yes**	2x*	-	10^{-4}
14	VerSe	**Yes**	3x*	-	10^{-4}
15	VerSe	**Yes**	4x*	VerSe_2x*	10^{-4}
16	VerSe	No	3x	DIV2K_GRAY_3x	10^{-5}
17	Verse	**Yes**	3x*	DIV2K_GRAY_3x*	10^{-5}

*Asymm? = "asymmetric upsampling module", SF = scale factor, LR = learning rate, * indicates asymmetric scaling factor.*

adjustment is made at epoch 200 where the learning rate is halved. During the training, the model that performed best on the validation set was saved along with the final model. Testing was performed on the best-performing model.

Models 1-3 are trained with the DIV2K set with symmetric scaling. These are meant to serve as a baseline from which to compare the performance of the symmetric models with results that are reported in the literature. Models 4-6 are trained with the DIV2K_GRAY set, again with symmetric scaling. These are meant to be compared directly to models 1-3 to check the relative performance of image SR between color (3 color channel) and grayscale (single color channel) natural images. Models 7-9 are also trained on the DIV2K_GRAY dataset, but these use the purely asymmetric upsampling module described in Section 4.1.1, and as such also use an asymmetric scaling factor. These are meant to provide an initial benchmark for the difficulty of purely asymmetric SR by allowing it to be compared to the symmetric models 4-6. Models 10-12 are trained on the custom VerSe dataset with symmetric scaling factors. These models can be compared to the symmetric DIV2K_GRAY models 4-6 to gauge the relative difficulty

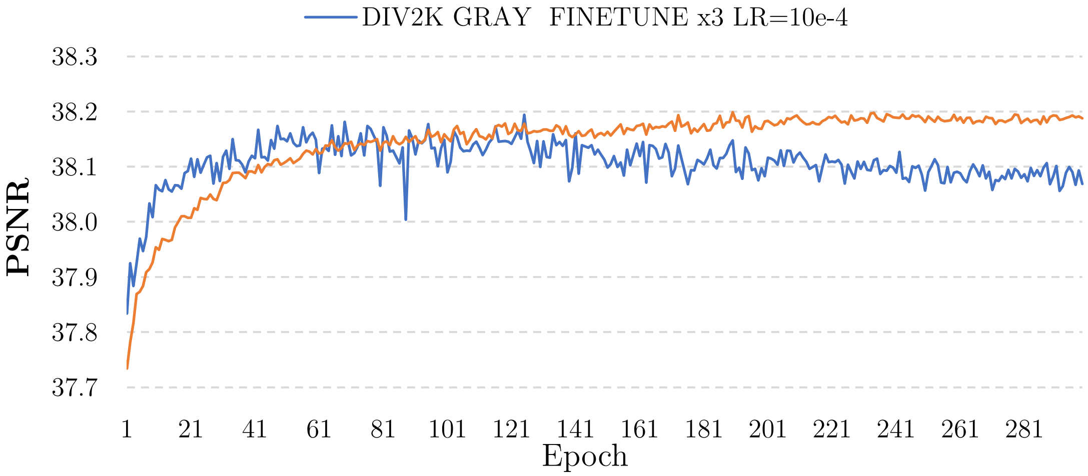


Figure 20. Validation scores for finetuning models on VerSe data (pre-trained on DIV2K_GRAY), showing the overfitting on this VerSe training set can be mitigated by decreasing the learning rate.

of the general SR task on the VerSe dataset. Models 13-15 are trained on the VerSe dataset but use the asymmetric upsampling module with asymmetric scaling factors. These are the most relevant models to the larger problem, as they will show qualitatively and quantitatively the SR reconstruction performance on under-sampled medical images of the spine – this task being the primary motivation for the asymmetric SR module. Finally, models 16 and 17 are trained on the VerSe dataset but are pre-trained on the DIV2K_GRAY dataset. This is to check if pre-training on DIV2K and finetuning for a specific task will confer any reconstruction performance benefit in either the asymmetric or symmetric upsampling case.

The models just described were all tested on holdout data from their own dataset, as described in Section 4.1.3. The DIV2K models were tested on *both* DIV2K and VerSe test data. This is to answer whether a model trained on natural image data will perform better, worse, or similarly to a model trained on task-specific data when used for inference on the task-specific test set. On the other hand, the models trained on VerSe data are only tested on VerSe data. "How an SR model trained on CT images of the spine will perform on diverse natural image data" is not a research question answered in this thesis. Based on imbalance of texture diversity between the two datasets (DIV2K having much more diverse texture), my expectation is that the performance would be sub-optimal. Finally, the two VerSe-finetuned models with DIV2K pre-training are also tested only on the VerSe test set. The reconstruction error from rescaling each test set with bicubic interpolation was calculated to provide a naïve baseline for SR performance of every model.

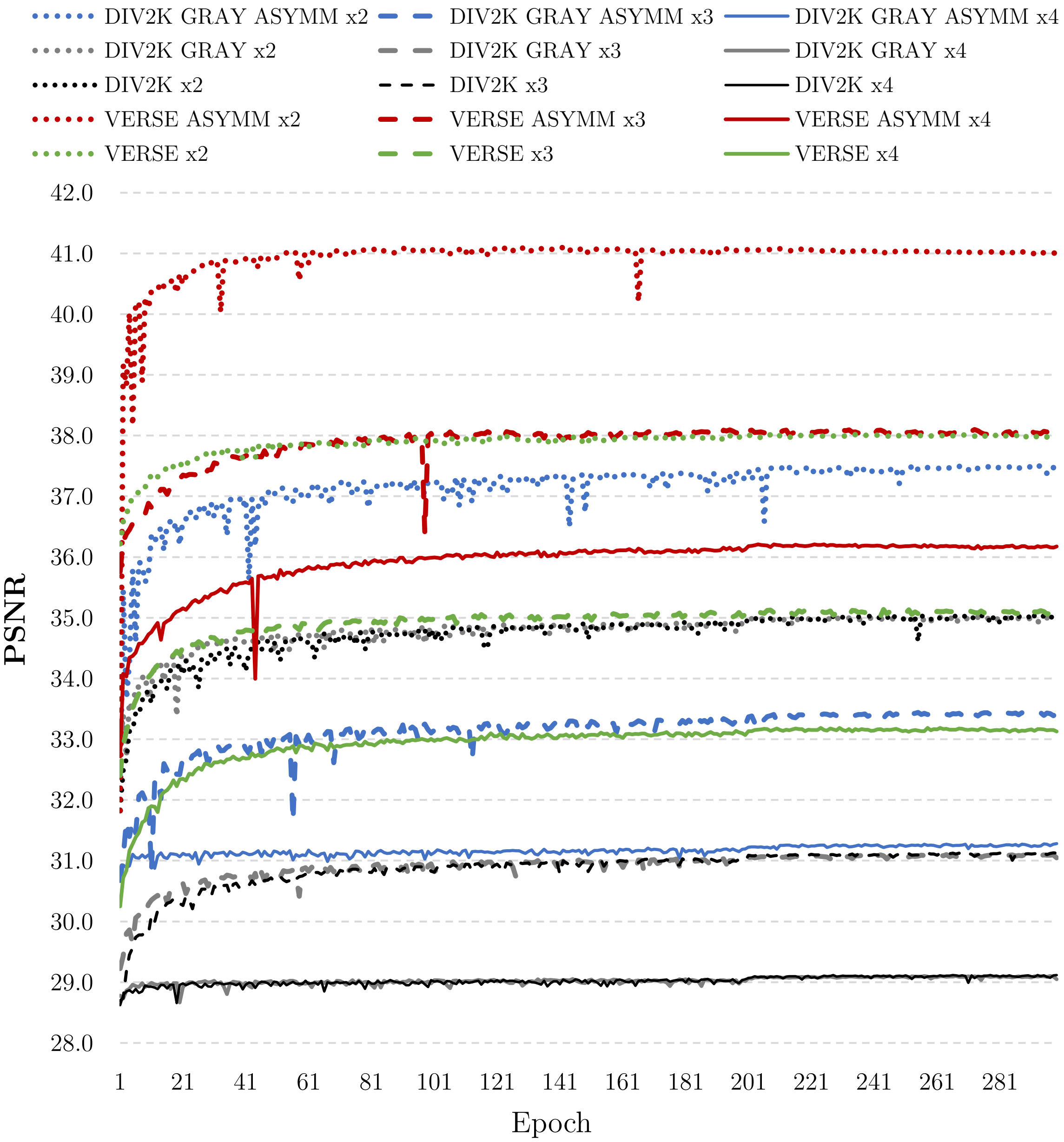


Figure 21. Validation scores during training of the generative hybrid SR models. Color-coded by training/validation dataset, 2x scaling factor models are dots, 3x are dashes, 4x are lines.

4.2 Results

4.2.1 Quantitative Results

The experimental results comparing SR reconstruction performance on the test sets are found in Table 5. I compare the DIV2K results to those of [102], who also test a modified SRResNet on the DIV2K validation set. They report scores of 34.40 dB (2x), 30.82 dB (3x), and 28.92 dB (4x) for SRResNet tested on holdout DIV2K validation

TABLE 5. TEST SET RECONSTRUCTION PERFORMANCE (PSNR)

		scale	Training Dataset			
			DIV2K	VERSE	P-VERSE	Bicubic
Test Dataset	DIV2K	x2	34.0884			30.8996
		x3	30.5493			28.1880
		x4	28.7131			26.6560
	DIV2K _GRAY	x2	33.997			30.9465
		x3	30.4528			28.2519
		x4	28.6448			26.7381
	DIV2K _GRAY	x2*	36.5112			33.3228
		x3*	32.7557			30.3424
		x4*	30.7876			28.6720
	VERSE	x2	38.3765	**38.7394**		31.1532
		x3	35.0802	35.9985	**36.0943**	28.6109
		x4	33.1451	**34.1897**		27.2185
	VERSE	x2*	41.4375	**41.7206**		34.6101
		x3*	38.0641	**38.8349**	38.7879	32.0183
		x4*	36.2027	**36.9373**		30.5153

** indicates asymmetric scale factor, P-VERSE means pre-trained on DIV2K. Units are dB PSNR.*

images. While the results from my symmetric models on this test set are slightly lower, the performance is comparable. This finding establishes a baseline: my model's architecture (when used in "symmetric upsampling mode") performs similarly to the modified SRResNet used in [102].

An interesting result is that the DIV2K and DIV2K_GRAY models saw nearly identical SR performance, with DIV2K_GRAY-trained models scoring consistently-yet-negligibly lower on the DIV2K validation subset that I used as a holdout test set. The asymmetric upsampling modules show good SR performance on both the DIV2K and VerSe test sets, greatly outperforming bicubic interpolation.

Scaling asymmetrically by a factor of 2x*, 3x*, or 4x* is obviously an easier problem than scaling symmetrically by the same factors. The VerSe dataset itself also appears to be an easier SR problem than DIV2K as measured by PSNR, evidenced by the VerSe validation and test scores greatly exceeding those of DIV2K for both the asymmetric and the symmetric SR models. This is likely due to the large number of near-black pixels that appear in the background of these medical images, forming large featureless dark areas that are particularly easy to super-resolve.

It was also found that training on the task-specific VerSe data did confer a reconstruction performance benefit on the VerSe test set as measured by PSNR. On the other hand, the DIV2K models finetuned on VerSe training data do not show a consistent performance benefit. The symmetric pretrained-on-DIV2K 3x model only slightly outperforms the VerSe-trained model, but the asymmetric pretrained model scores less than the task-specific VerSe-trained model. This suggests that DIV2K pre-training for the task-specific VerSe data does not substantially improve task-specific performance.

4.2.2 Qualitative Results

Qualitative results for the asymmetric SR models are provided for the DIV2K_GRAY and VerSe test sets. In Figure 22 some SR reconstruction results for the 2x*, 3x*, and 4x* asymmetric models are shown below the bicubic interpolations of the same scaling factors. The model seems to be good at reconstructing sharp elongated edges and is perceptually superior to bicubic interpolation which tends to smooth high frequency features such as edges. Figure 23 shows the performance of the asymmetric VerSe-trained models on the VerSe test set. On the VerSe test set the SR models are able to resolve the boundaries between the lumbar, thoracic, and cervical vertebrae. I also applied the asymmetric 3x* SR model to the under-sampled HSCT patient volumes described in Chapter 3. With this unseen dataset, the SR convnet models seem to have more trouble generating boundary pixels between the individual vertebrae. This can be seen in the top example of Figure 24, where the lumbar vertebrae are super-resolved with good separation, but the thoracic vertebrae are not. This is a qualitatively different result than what is shown by the SR results on the VerSe test set, where the boundaries between the thoracic and even the cervical vertebrae were resolved clearly. Some larger-format examples from the HSCT patient dataset used with the VerSe-trained asymmetric 3x* SR convnet are provided in Appendix B.

4.3 Discussion

Motivated by the task of upscaling under-sampled medical images, I proposed an asymmetric upsampling module that “drops in” to most modern state-of-the-art convnets for image super-resolution. The module is compatible with SR convnets that use learned sub-pixel [100] upsampling after features are learned in the low-resolution space, such as EDSR [102] and RCAN [35]. The module is a simple design that uses transposed convolution with asymmetric stride and asymmetric kernels to perform the asymmetric upsampling. To my knowledge, this is the first convnet built specifically to perform

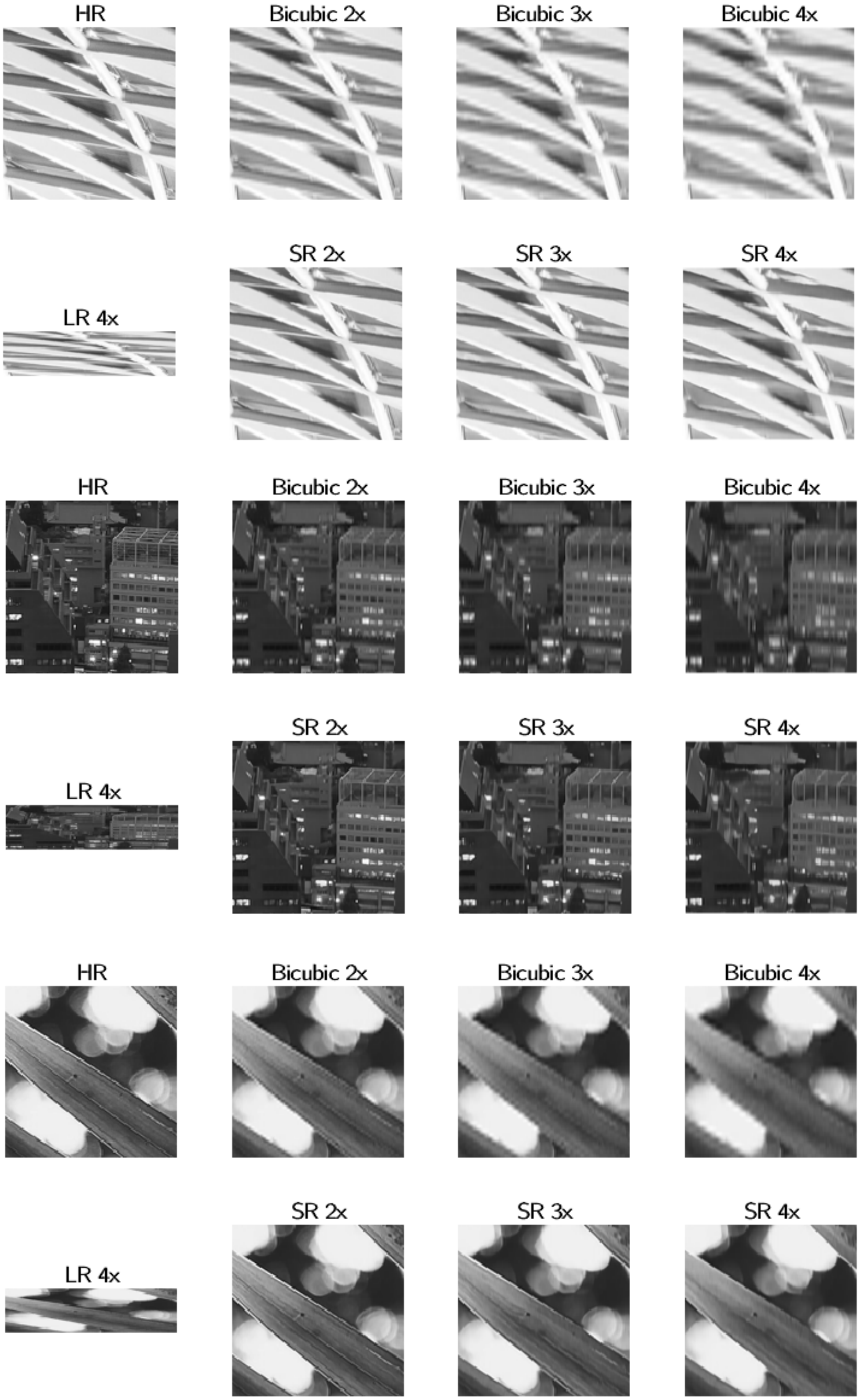


Figure 22. Asymmetric SR at 2x*, 3x*, and 4x* scaling factors trained and tested on the DIV2K_GRAY dataset.

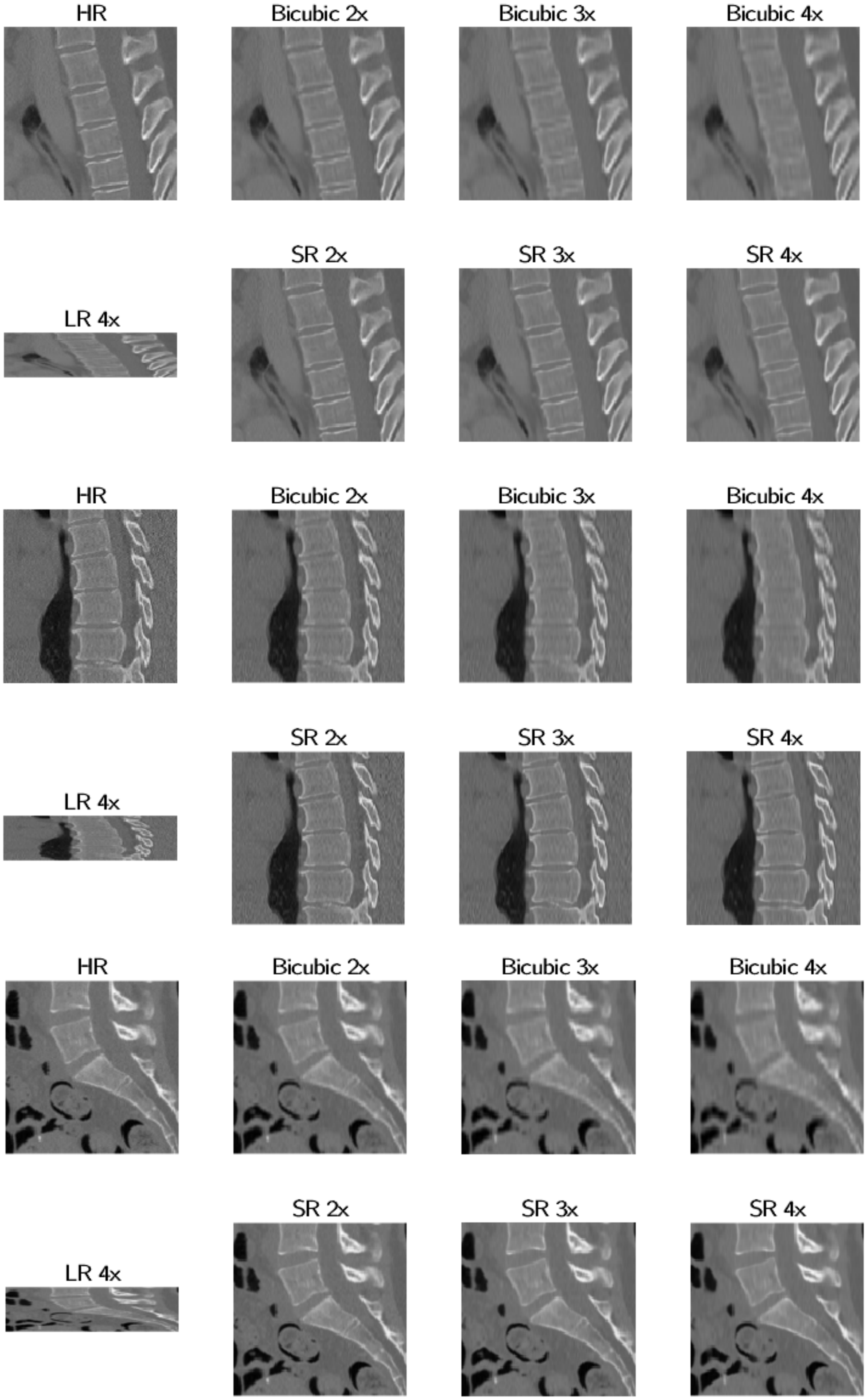


Figure 23. Asymmetric SR at 2x*, 3x*, and 4x* scaling factors trained and tested on the VerSe dataset.

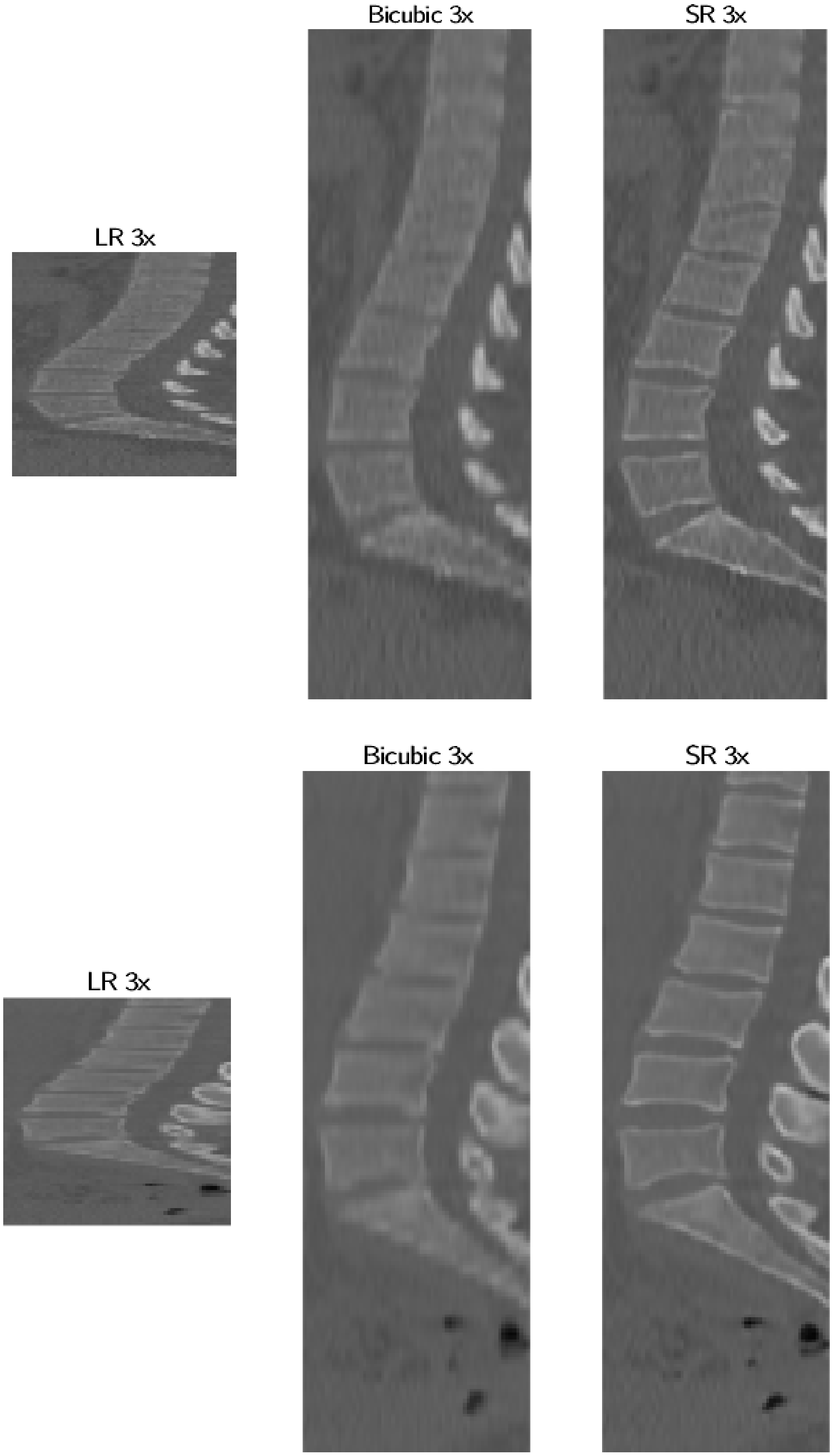


Figure 24. Asymmetric SR at 3x* on the HSCT patient dataset, trained on the VerSe dataset.

purely asymmetric super-resolution. The work from [21] provides a convnet for general asymmetric SR but they do not test the model on the *purely* asymmetric case where one dimension is held to unity while the other is scaled. I trained, validated, and tested super-resolution models on the DIV2K dataset as well as a custom-made task-specific medical imaging dataset. This custom dataset was built from the publicly available VerSe [29] CT volumes, from which I took high-resolution sagittal CT images of the spine. It was found that the symmetric models performed nearly identically on the 3-channel DIV2K dataset and the single-channel DIV2K_GRAY dataset. The asymmetric performance on DIV2K_GRAY is good, with the asymmetric 4x* model having similar reconstruction performance to the symmetric $3x$ model. The VerSe-trained models performed better on the VerSe test set, outperforming the DIV2K-trained models and the VerSe-finetuned models pre-trained on DIV2K. The asymmetric VerSe-trained models on the custom VerSe test set attained PSNR scores of 41.72 dB for the asymmetric 2x* model, 38.83 dB for the asymmetric 3x* model, and 36.93 dB for the asymmetric 4x* model.

Qualitative performance on CT volumes from the HSCT patient dataset described in Chapter 3 is not as good as seen on the VerSe test set. This may be explained by a number of factors. Foremost is that the VerSe dataset I created contains low-resolution images down-sampled from high-resolution images. The bicubic downsampling procedure I used in creating the low-resolution images for the VerSe training set may not be a reasonable approximation of the in-situ under-sampling of a CT volume. It is possible that training SR convnet models with low-resolution images generated from a more representative downsampling procedure may increase the performance on the HSCT patient dataset (nearest-neighbor may be a reasonable choice). Of course, there are differences in the CT imaging protocols used for capturing the CT image volumes of the VerSe and HSCT patient datasets. These technical differences may also play a role in the qualitative performance discrepancy.

Asymmetric super-resolution convnets prove to be a good method to accurately upscale under-sampled medical image data. An obvious next step for this SR use case is a generative SR convnet architecture for 3D LR/HR training patches with the asymmetric implementation in mind. Pham et al. present a fully-3D SR convnet (albeit for symmetric upscaling) in [117], but this is a very shallow network built as a 3D extension of SRCNN [98] and lacks the representative capability of the more recent and much deeper 2D SR convnets. The main challenge with the 3D SR approach is that making a much deeper network will be computationally very expensive. EDSR is already very large at 43 million parameters, and it is just a 2D image convnet [102]. Another area to improve is the medical imaging training set. The custom VerSe training set that I built is not particularly diverse, and I believe generalization performance of the SR convnets on the medical image upscaling task could be improved by expanding and curating high-resolution medical images to be used as a general benchmark, much like DIV2K has become the standard for training on natural image data. Lastly, the sub-pixel convolutional layer would be a more efficient alternative to the transposed convolution used in my purely asymmetric upsampling module. An implementation of asymmetric sub-pixel upsampling would be a useful tool to add to the mainstream deep learning frameworks.

Chapter 5
Vertebral Body Instance Segmentation

The convnet-based super-resolution methods described in Chapter 4 fail to consistently resolve the boundaries between individual vertebrae in the HSCT patient dataset based on the CT data alone, motivating other methods to complete the vertebral body instance segmentation task. In many cases the FLT-PET data allows the detection of most vertebral boundaries by simple visual inspection of sagittal slices around the spine region. This is exemplified by Figure 25. In this chapter I present a simple algorithm robustly detects vertebral boundaries from the under-sampled post-HSCT day-28 FLT-PET image volumes that were captured alongside the CT volumes. Somewhat ironically, the FLT-PET data, which is of lower resolution than the CT data (and thus would generally be considered less useful for localization) will be used here to localize anatomical structures with improved accuracy. The detected vertebral boundaries are used to individually segment the vertebral bodies from the vertebral body class mask found by the methods of Chapter 3. Note that Nguyen et. al use a Kalman filter for this task in [10, 11]. Although the approach I used is relatively simple, it is robust.

Later in this chapter I use the individual vertebral body segmentations to extract standardized uptake value (SUV) [26] measurements from the FLT-PET volumes. I do the same for the pelvis and sternum bone structure segmentations, and I show an automatic method for visualizing the FLT-PET data from these various regions of interest (ROIs) by constructing isosurfaces from the cumulative distribution of FLT-PET values in a given ROI volume.

5.1 Methods

5.1.1 FLT-PET Image/CT Mask Registration

The vertebral boundary detection algorithm operates on the FLT-PET data, and the vertebral body segmentation masks generated in Chapter 3 are used for an initial masking of the FLT-PET region of interest. The first challenge is to reconcile the coordinate system differences between the segmentation masks and FLT-PET volumes. As mentioned in Section 3.1.1, the FLT-PET volumes have an axial slice size of only

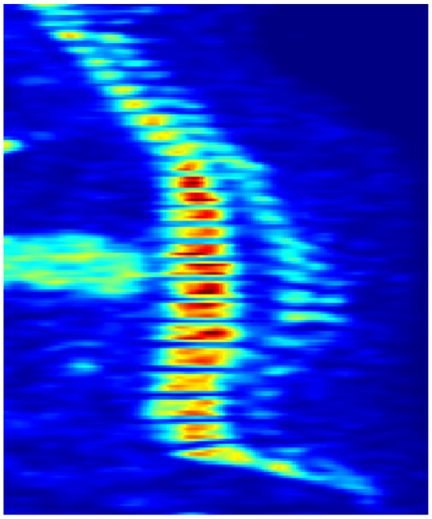


Figure 25. Sagittal view FLT-PET image of a HSCT patient on the 28th day post-transplant. The vertebral boundaries are clearly defined by local differences in cell proliferation activity.

144×144 pixels compared to the CT volume's (and therefore a segmentation mask volume's) axial slice pixel-resolution of 512×512. Bicubic interpolation is used to rescale each axial slice in the FLT-PET volume to 512×512 pixels to match the axial slice size of the mask image volumes. Note that this up-scaling interpolation is simply for detection of the vertebral boundaries and is not recommended for analysis of the PET data itself for reasons outlined in Section 5.2.2. After the interpolation, the FLT-PET volume has a pixel-resolution of $512 \times 512 \times H_{\text{PET}}$ and the and the segmentation mask volume has a pixel-resolution of $512 \times 512 \times H_{\text{CT}}$, where H_{PET} and H_{CT} represent the pixel size in the axial direction of the FLT-PET and CT data, and in general $H_{\text{PET}} \neq H_{\text{CT}}$. The joint CT-PET scanner provides a global "z" coordinate (along the axial direction) for each axial slice, allowing the segmentation masks and the PET data to be axially registered by interpolation. The *interp3()* MATLAB function was used for this operation, which samples image data from one coordinate grid to another using a selectable interpolation method. There are two ways to proceed with this (second) interpolation: interpolate PET data from the CT/mask coordinate grid or interpolate mask data from the PET coordinate grid. Interpolating a new segmentation mask from the PET coordinate grid preserves the additional resolution provided by the FLT-PET volumes (as mentioned in Section 3.1.1, the FLT-PET data was captured at a higher resolution than the CT data, or $H_{\text{PET}} > H_{\text{CT}}$). Additionally, interpolating the segmentation mask to the FLT-PET coordinate grid detects the vertebral boundaries in the original PET coordinate system, which is useful for analyzing the PET data. For these reasons, interpolating the CT mask from the PET coordinate system is the best of the two options, and this is how I proceed, using the *interp3()* MATLAB function with the “linear” interpolation method. Linear interpolation will be adequate for resampling the vertebral body mask since the object shape change between neighboring axial slices is minor. Interpolated mask values falling between zero and one are simply thresholded. After the interpolations, we are left with a CT mask volume and a FLT-PET image volume of equal $512 \times 512 \times H_{\mathrm{PET}}$ pixel-resolution.

5.1.2 Morphological Preprocessing

Before masking the FLT-PET data with the vertebral body segmentation mask, some preprocessing is performed on the mask to regularize the shape. As shown in Chapter 3, sometimes the U-Net will be able to discern the vertebral boundary between the wider-spaced lumbar vertebrae, creating a discontinuous vertebral body mask. Since the boundary detection is now being performed in the FLT-PET modality, these “gaps” need to be filled so that the mask covers the entirety of the FLT-PET data in the vertebral body column region-of-interest. To accomplish this a simple 3D morphological closing filter is applied (with a small spherical structuring element of radius 1). With the vertebral body column now contiguous, a connected-components method is used to select and extract the largest contiguous object in the CT mask volume, removing any “islands” of false-positive voxels from prediction masks, such as those shown in Figure 26.

5.1.3 Dimensionality Reduction and Boundary Detection

After regularizing the vertebral body column mask shape the mask is applied to the FLT-PET volume data. At this point a reasonable next step is to reduce the dimensionality of the problem by averaging the FLT-PET intensities of the masked pixels for each axial slice according to

$$I_{avg}(i) = \frac{1}{N_i} \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} V_{PET}(j,k,i)\,, \tag{19}$$

where $I_{avg}(i)$ is the mean FLT-PET intensity of the i^{th} axial slice in the masked FLT-PET volume image V_{PET}. N_i is the number of vertebral body object pixels in the i^{th} slice. The result is a one-dimensional signal I_{avg}, from which the boundaries between individual vertebrae are easily identifiable as valleys. This axial-plane averaging approach to reducing the problem to one dimension is also used by Nguyen et. al in [10, 11].

Inverting the sign of the signal I_{avg} allows the peak-detecting MATLAB function *findpeaks()* to locate the valleys in the original signal. Simply using the function with no

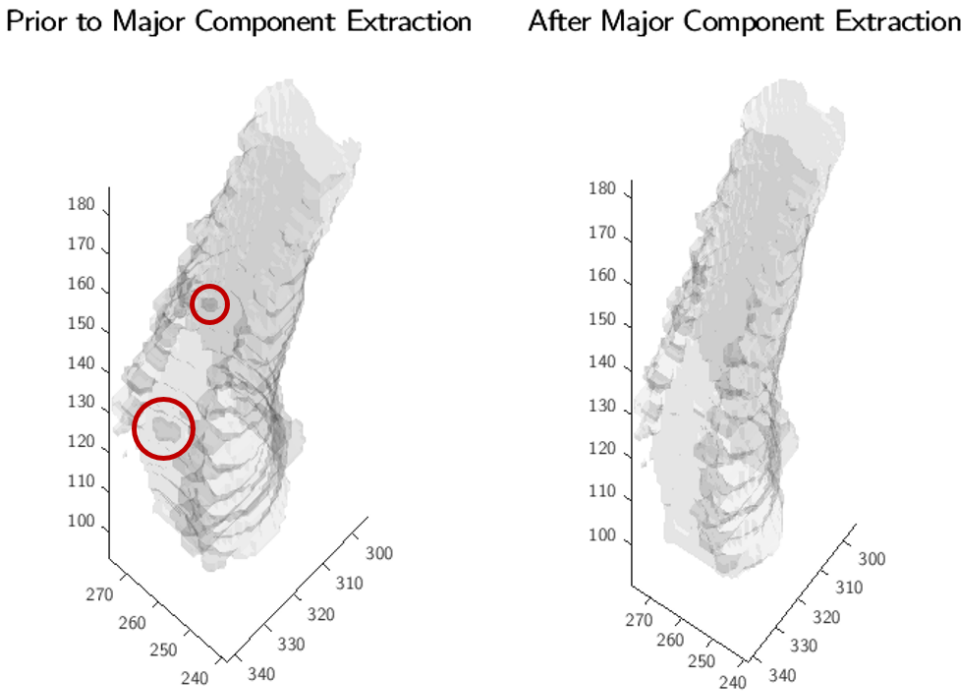


Figure 26. Major object extraction removes small groups of false-positive voxels (circled in red).

qualification method for the peaks yields “false” peaks due to small fluctuations in hematopoietic cell proliferation activity within the vertebral bodies. This can be seen in the central graph of Figure 27. These false peaks can be filtered out by imposing a qualification test, such as setting a minimum peak height or a minimum distance between peaks. Qualifying peaks by setting a minimum height is challenging because the expected height of the peaks representing the vertebral boundaries can fluctuate greatly from patient to patient and, for any given patient, from vertebrae to vertebrae. Qualifying peaks by setting a minimum distance is decidedly less challenging. While the expected minimum distance between “correct” peaks (corresponding to ground-truth boundary indicies) also varies from patient to patient, the variance is much smaller. Setting a minimum peak distance of “5” removes the false peaks. However, the minimum distance between peaks also varies from vertebrae to vertebrae for a given patient. This can be can be seen in the bottom graph of Figure 27, where some vertebral boundaries in the thoracic and cervical vertebrae (higher up the spine) do not qualify as peaks because they are within 5 axial slice indices of another stronger peak.

Since the MATLAB function *findpeaks()* does not allow for a linearly decreasing minimum peak distance, I write my own detection algorithm for this specific use case. The peaks detected by the MATLAB function are used to find a prior for the span of

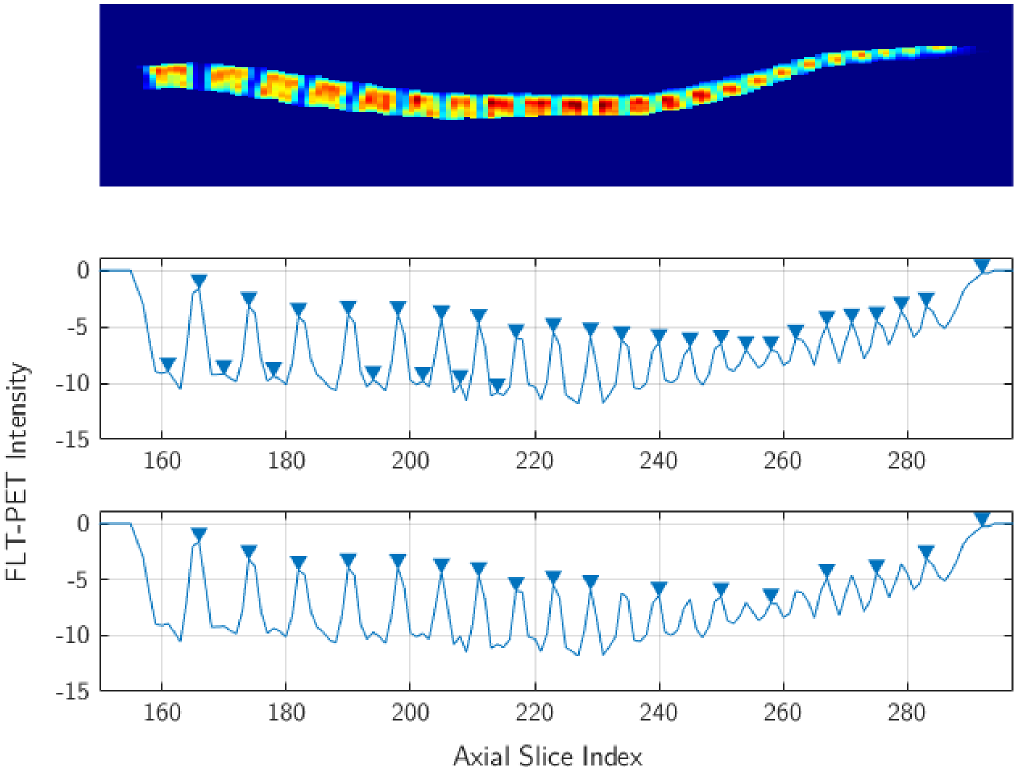


Figure 27. A sagittal slice of the masked FLT-PET data (top). Result from running peak detection algorithm with no qualification parameters on the inverted I_{avg} signal (center). Result from running peak detection algorithm with an established minimum peak distance of "5" (bottom).

the L4 vertebrae in the axial direction, d_{L4}, by taking the difference of the indices of the first two detected peaks (which can always be detected robustly). The index of the axial slice representing the beginning of the L5 vertebrae is taken to be the index of the first nonzero value in I_{avg}. Then, from the index between the L4 and L3 vertebrae, the next boundary is found by finding the minimum within the next d_{L4} values of I_{avg}, excluding the nearest $\left\lfloor \frac{d_{\mathrm{L4}}}{3} \right\rfloor$ indices which physically cannot be the index of the next vertebral boundary, simply due to their proximity. This "exclude nearest" design mitigates a failure mode where two vertebral boundaries can be detected between the same two vertebrae, depending on the average intensity values on the "other side" of the next vertebrae. This failure mode is typically prone to occur between the lumbar vertebrae, which have wider boundary regions spanning two or three axial slices with lower I_{avg} values. The value of $\left\lfloor \frac{d_{\mathrm{L4}}}{3} \right\rfloor$ is chosen from analysis of the HSCT patient dataset; no vertebral boundaries are ever detected within $\left\lfloor \frac{d_{\mathrm{L4}}}{3} \right\rfloor$ indices of the previous vertebral boundary. This is made evident in Figure 28, where all the vertebra span at least $\frac{1}{3}d_{\mathrm{L4}}$. So, when i is the last detected boundary index, the search for the next boundary is limited to the window of

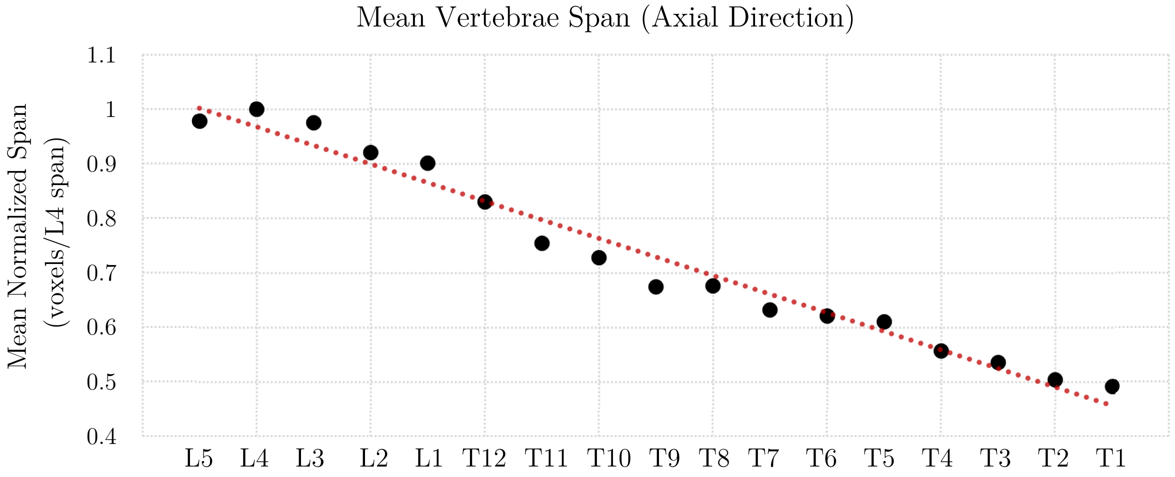


Figure 28. Mean span of each vertebrae in the HSCT dataset normalized to the initial vertebrae span prior (L4).

values between $I_{avg}(i + \lfloor \frac{d_{\text{L4}}}{3} \rfloor)$ and $I_{avg}(i + d_{\text{L4}})$, which only contains a single vertebral boundary region. Also shown in Figure 28 is that the expected vertebral span decreases for vertebrae higher in the spine. Using the initial vertebral span prior d_{L4} for each vertebral boundary search is not feasible, as beginning near the mid- and upper-thoracic regions d_{L4} might span multiple vertebral bodies. So, d_{L4} is modified after detecting the end of the T12 vertebrae to a new value $d_{\text{T12}} = \lceil \frac{4}{5} d_{\text{L4}} \rceil$, modified again at the T7 vertebrae to a value $d_{\text{T7}} = \lceil \frac{3}{4} d_{\text{L4}} \rceil$, and last modified at the T4 vertebrae to $d_{\text{T4}} = \lceil \frac{3}{5} d_{\text{L4}} \rceil$. These changes to the vertebral spanning prior were determined by the analysis of the vertebral boundaries on the HSCT patient dataset shown in Figure 28. The detection of vertebral boundaries continues iteratively in the method described above until all lumbar and thoracic vertebral bodies have been bounded. The cervical vertebral bodies are omitted due to inconsistency of the algorithm in that region. The instance segmentation of the lumbar and thoracic vertebral bodies is then a simple matter of slicing the 3D vertebral body mask at the detected boundary indices.

5.2 Results

Results from the instance segmentation algorithm are very good. Table 6 shows the magnitude of the difference between the manual labelling of vertebral boundaries and the boundary indices detected by the algorithm. In most of the cases where differences appears in Table 6, the algorithm is working as intended. Ground-truth labelling was conducted "by eye", using only a single sagittal view of the FLT-PET data, whereas the algorithm is computing the mean value of the masked 2D FLT-PET data in each axial slice. Put another way, the algorithm is considering more information than was available

Table 6. Magnitude Difference Between Manual and Algorithmic Boundary Index Detection Methods

	init.	L5	L4	L3	L2	L1	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	**RMSE**
p1	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	0.236
p2	1	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	1	-	0.408
p3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.236
p4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.000
p5	1	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	0.333
p6	1	1	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.408
p7	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	0.236
p8	1	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	0.333
p9	1	1	1	1	-	-	-	-	-	-	-	-	-	1	-	1	-	-	0.577
p10	-	-	1	-	-	1	-	-	-	-	-	-	-	-	1	-	-	-	0.408
p11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.236
p12	1	1	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	0.408
p13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.000
p14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	0.236
p15	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0.408
p16	1	-	1	1	-	-	-	-	-	-	-	1	-	-	-	-	-	-	0.471
p17	1	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	1	1	0.471
p18	-	-	1	-	-	-	-	-	-	-	-	-	-	-	1	-	-	3	0.782
p19	1	-	-	-	-	-	1	1	-	-	-	-	-	-	1	-	-	-	0.471
p20	-	-	1	1	1	-	-	-	-	-	-	-	-	-	-	-	1	-	0.471
p21	1	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	0.333
p22	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	1	1	1	0.471

*“-“ indicates a magnitude difference of zero for the boundary. “*p_*” indicates the patient number in the anonymized HSCT dataset.*

when the ground-truth boundary indicies were manually determined. Additionally, it is hard for a human annotator to perceptually determine a minimum between two axial slices when both appear to be equal in mean intensity, and this situation arose frequently during ground-truth labelling. The algorithm does not have this problem. So, where “1’s” appear in Table 6, it is often because the boundary detected by the algorithm is simply uses pixel information from each entire axial slice instead of a single sagittal view. Table 6 also shows the root mean squared error (RMSE) of the algorithmically detected boundaries for each patient volume, given by [118]

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i}^{N}\left(\theta_i - \hat{\theta}_i\right)^2} \quad , \tag{20}$$

where θ_i and $\hat{\theta}_i$ are the ground-truth and algorithmically derived boundary indexes at each index i, and N is the total number of vertebral boundaries detected. Notably, the method I implemented produces lower RMSE for each patient than the Kalman filter method shown by [10], in some cases by large margins. However, it must be noted that in [10] the authors include the cervical vertebrae, which I omit. It likely that most of their reported RMSE comes from this region, so the scores cannot be directly compared [10]. As shown in Table 6, the algorithm produces only one obvious mistake at the T1 vertebrae for the patient 17 image volume. An example of the resulting instance segmentation is shown in Figure 29, with examples from the rest of the dataset appearing in Appendix C.

5.2.1 FLT-PET Visualization

Since the CT mask volume has been registered to the interpolated FLT-PET volume, extracting cell proliferation data for a vertebral body instance is as a simple task of applying the vertebral body mask. I developed a script for automatic visualizion of this data which determines isosurfaces of FLT-PET intensity levels to create a "heatmap-like" 3D view of cell proliferation within a given bone structure ROI. Following the same interpolation procedure as was described in Section 5.1.1, the CT-derived masks are intepolated to the PET scan locations, and the PET data is upscaled in the axial plane to match the axial dimensions of the segmentation mask volume. FLT-PET isosurface values are selected by constucting discrete cumulative distribution functions from histograms of the FLT-PET intensity values contained within the ROIs. Setting isosurfaces at the approximate 50th, 85th, and 98th percentiles from the FLT-PET cumulative distribution creates a good visualization of cell proliferation activity. The isosurfaces are interpolated in MATLAB using the built-in *isosurface()* function. The PET activity on the approximate 28th day post-transplant is shown in Figure 30 and Figure 31 for the vertebral body and pelvis ROI's, respectively. Visualizations from the rest of the dataset are provided in Appendix D.

5.2.2 Standardized Uptake Value Measurement (FLT radiotracer)

In addition to visualization, the segmentations were used to perform a calculation of the standardized uptake value (SUV) within the ROIs. For this calculation, instead of

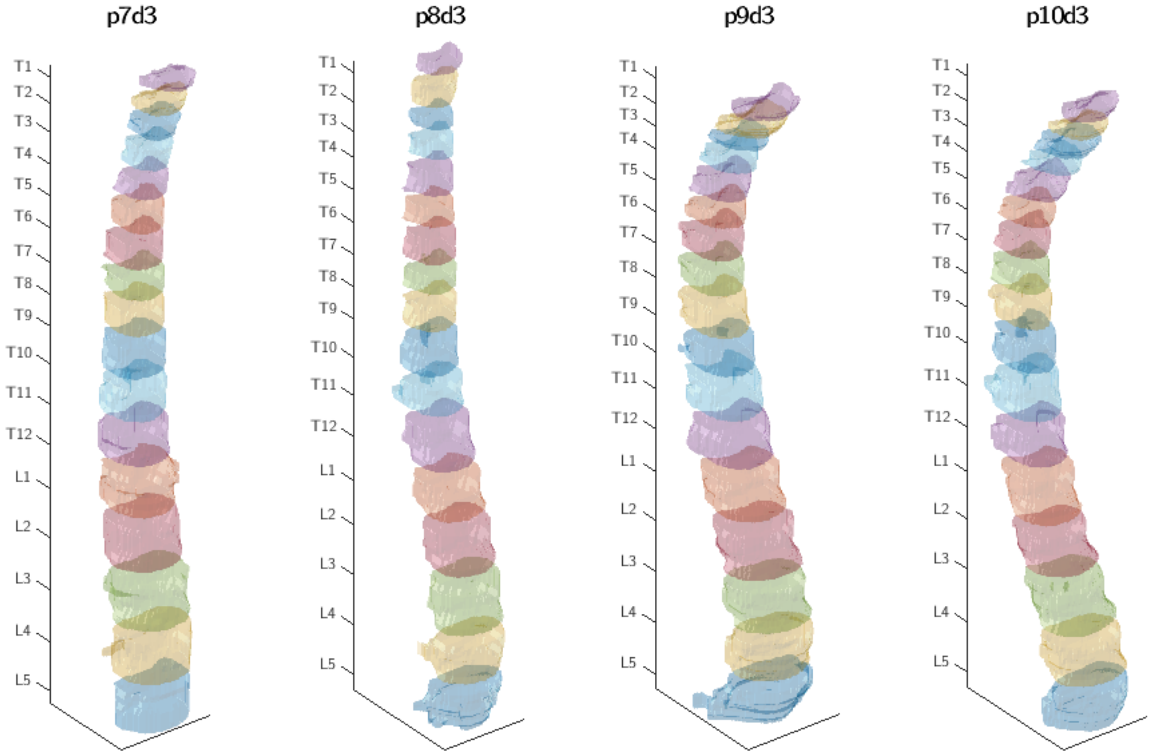


Figure 29. Vertebral-body instance segmentation results from the methods described. The code "pXd3" indicates the patient number and scan index (where "d3" is approximately the 28th day post-HSCT).

upsampling the FLT-PET data (which is already in units of SUV for this HSCT patient dataset), the segmentation masks were downscaled to match the PET dimensions in the axial plane. The choice to downscale the segmentation masks rather than upscale the PET volumes was made for a few reasons. First, since SUV is defined as a normalized "radioactivity per volume" (see Section 2.1.2), if the voxel dimensions are rescaled, the voxel intensities need to be rescaled as well. Put in other words, rescaling the PET data when it is already in units of SUV changes the units to "SUV times a constant". While a ratio of voxel sizes may provide a good estimate for this constant, the interpolation still may not conserve the SUV in a volume, depending on the interpolation method. This would need to be investigated. Second, interpolation could cause high-intensity voxels to propagate outward, and these interpolated voxels may cross the edges of the segmented bone structure ROIs (generated in Chapter 3) where they would not be accumulated in the SUV measurement as they should be. While it may be possible to solve these issues, simply choosing not to rescale the PET data (and instead choosing to downscale the segmentation masks) nullifies each of them.

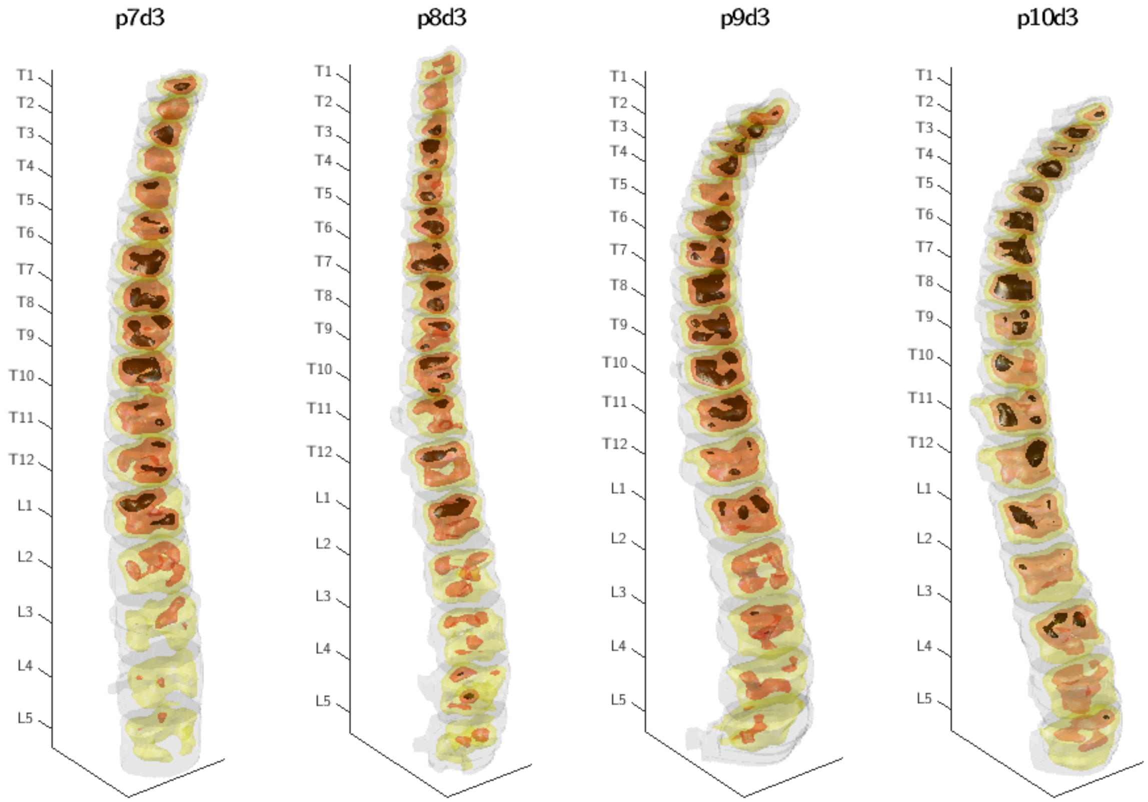


Figure 30. FLT-PET visualization of vertebral body ROIs via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values, orange surfaces represent the 85th percentile, and red represents the 98th percentile. “pXd3” indicates the patient number and scan index in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

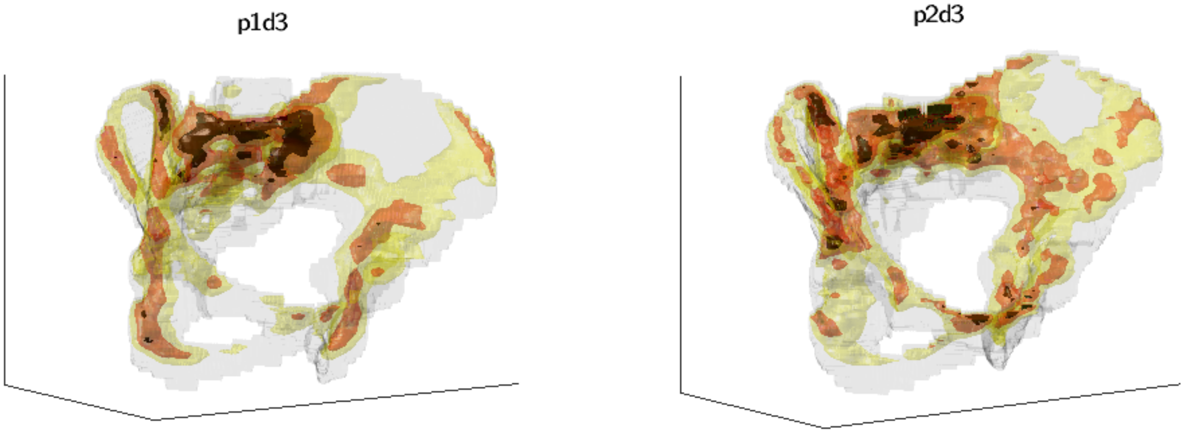


Figure 31. FLT-PET visualization of pelvis ROIs via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values, orange surfaces represent the 85th percentile, and red represents the 98th percentile. “pXd3” indicates the patient number and scan index in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

After downscaling the mask volumes, they are dilated with a small spherical structing element of radius one. This dilation is to account for the fact that the downscaling operation may cause the unwanted exclusion of high-intensity PET voxels near the edge of the segmentation that should be included in the SUV calculation. After the dilation, calculating the SUV of an ROI is a simple task of masking the PET data with a particular bone structure mask and adding up all the voxel intensities therein. By these methods, the SUV was computed for the vertebral body column, pelvis, and sternum ROIs. The same method was used to compute the SUV for the individual vertebral bodies, with one procedural difference: instead of a dilation with the 3D spherical structuring element, the volume is dilated with a 2D cross structuring element applied on the axial plane. The reason for this deviation in method is to prevent double-counting of the voxels between vertebral bodies. Due to the low resolution of the HSCT patient dataset along the axial dimension, the boundaries between the vertebrae are often only one voxel (or less) thick; a dilation via a spherical structuring element applied to individual vertebral bodies would span this gap, but dilation via the cross structuring element applied to the axial plane does not.

The vertebral body ROIs in the lumbar and thoracic vertebrae exhibited a mean SUV of 45,932 on the 28th day post-HSCT across the patient dataset. The SUV calculation within each vertebra across all patients (on the 28th day post-transplant) is shown in the boxplot Figure 32. Similar data for the pelvis and sternum ROIs are shown in Figure 33 across all three imaging days. On the 28th day post-HSCT, the mean SUV for the pelvis ROI was found to be 62,485 across the HSCT patient dataset, and the mean SUV for the sternum ROI was found to be 5,043. The complete tables showing SUV measurements for each patient and ROI are shown in Appendix E. A close examination of Table 8 in Appendix E reveals the mean SUV for the vertebral body column object class across the HSCT patient dataset is 54,051. This value is greater than the mean of the combined individual vertebral bodies for two reasons. First, the vertebral body column mask includes some PET data from the cervical vertebrae that is truncated in the instance segmentation procedure due to the inability of the vertebral boundary detection algorithm to function properly in that region of the spine for the under-sampled volumes in the HSCT patient dataset. Second, the vertebral body column segmentation includes the vertebral boundary regions removed by the instance segmentation, and these boundary regions contain low-intensity PET values which are aggregated in the SUV calculation.

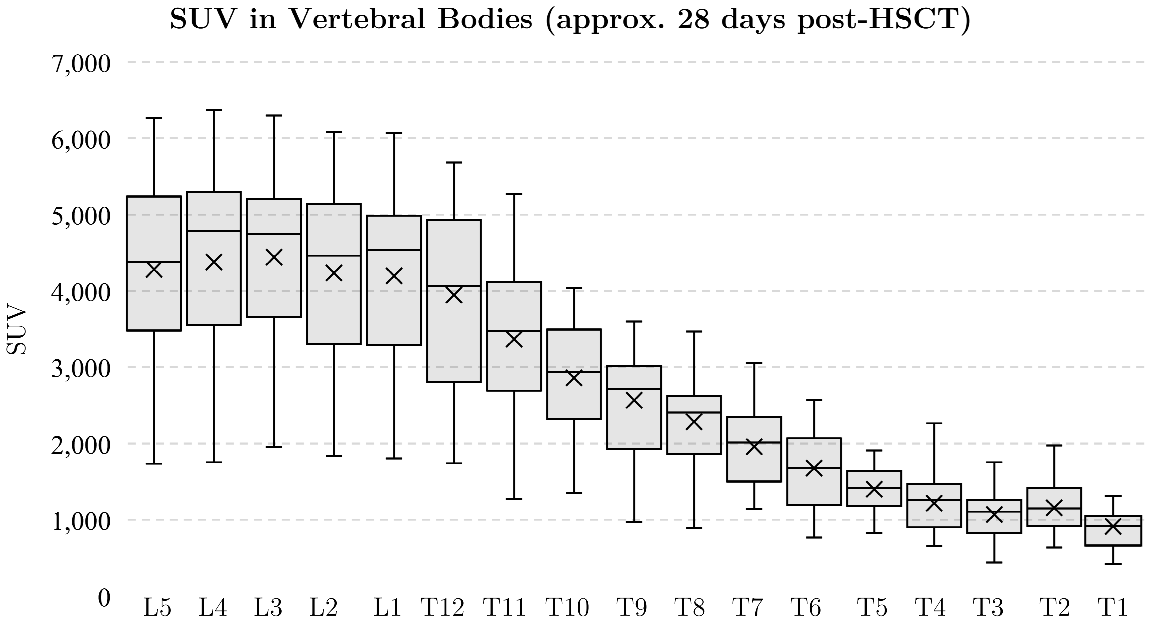


Figure 32. SUV for the vertebral body instances in the lumbar and thoracic spine. "×" indicates the mean value. Whiskers extend to largest/smallest value within 1.5 times the interquartile range.

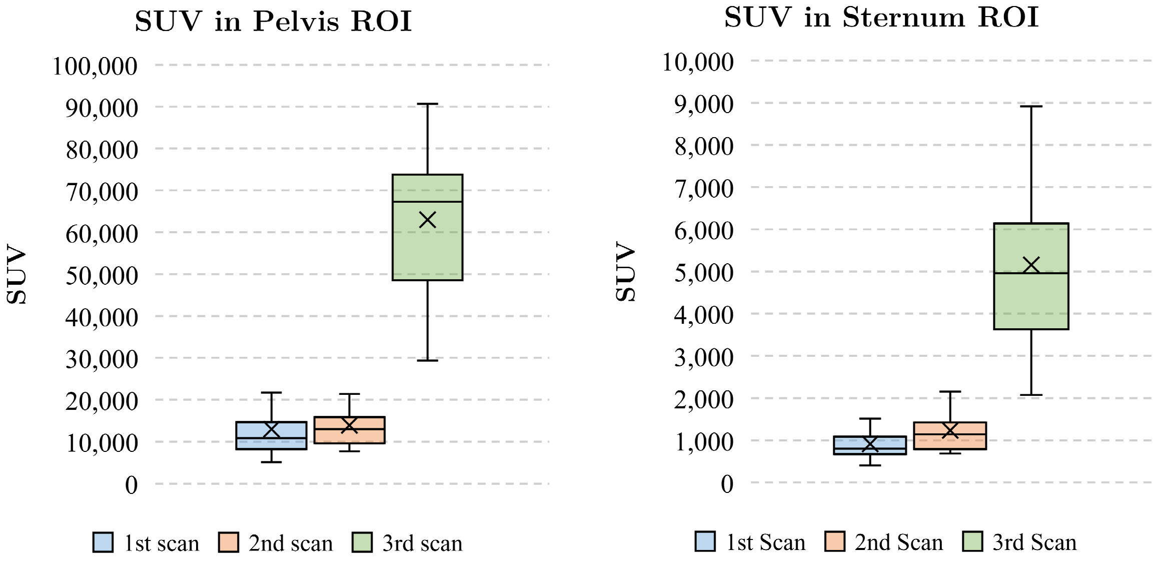


Figure 33. SUV of pelvis and sternum ROIs across the 3 imaging days. 1st scan is the day before transplant (ablated), 2nd scan is between 5 and 9 days post-HSCT, 3rd scan is 28 days post-HSCT. "×" indicates the mean value. Whiskers extend to largest/smallest value within 1.5 times the interquartile range.

5.3 Discussion

The vertebral body instance segmentation method described in this chapter is effective for dual-modality CT/FLT-PET image volumes (under-sampled in the axial/scanning dimension) of HSCT patients around the 28th day post-transplant. The algorithm is simple and robustly detects vertebral boundaries that are undetectable by humans and algorithms in the CT modality, and achieves lower root mean square error (RMSE) detection accuracy than the Kalman filter method of [10] for each patient in the HSCT test set, but only on the subset of the vertebral column containing the lumbar and thoracic vertebrae. Combining the detected vertebral boundaries with the vertebral body class segmentation masks predicted by the U-Nets of Chapter 3 allows the quantitative and qualitative analysis of FLT-PET data from specific vertebral body ROIs. However, there are some limitations. The method relies on FLT-PET imaging of stem cell proliferation in patients post-HSCT, which is currently an uncommon imaging technique. It also requires that the image volumes contain enough cell proliferation activity in the vertebral bodies that they can be bounded, as is the case with the 28th-day FLT-PET image volumes. Additionally, the method assumes that the vertebrae can be accurately split along axial slices from the image volume. This is often a good assumption, but not always. Some vertebral boundaries, particularly those in the upper-thoracic and cervical region in some patients, would be more accurately split by taking an oblique cut with respect to the axial plane. Another assumption is made in the visualization component of this chapter that bicubic interpolation of the FLT-PET data is a sufficient up-scaled visual representation of the original data. This is likely a reasonable assumption, but it would need to be critically examined before clinical use. Lastly, the algorithm used in this chapter is designed to work on the under-sampled image data of the HSCT patient dataset. The "minimum peak distance" parameter used to select the initial L4 vertebral body span prior from the MATLAB function *findpeaks()* would need to be adjusted (increased) for image data obtained from a higher axial sampling frequency. However, it should be noted that with such well-sampled image data the vertebral boundaries would likely be able to be detected directly in the CT modality using a convnet-based segmentation method and individually-labelled vertebral body training data. In other words, well-sampled CT volumes would negate the need for the instance segmentation algorithm of this chapter. Still, for the analysis of stem cell proliferation in under-sampled CT/FLT-PET image volumes, these methods are effective.

Chapter 6
Conclusion

The novel FLT radiotracer paired with PET imaging provides an unprecedented ability to examine post-HSCT cell proliferation in the entirety of the body's bone marrow compartments [6]. Compared to the traditional method of examining a patient's bone marrow compartments for hematopoietic activity – targeted invasive biopsy – FLT-PET imaging is an informational boon [8]. Typically, examination of this FLT-PET data is a time-consuming process where expert physicians draw and analyze regions of interest by hand. The drawing of ROIs is undertaken in 2D, or for 3D volumes, in 2D slice-by-slice. Automatic instance segmentation of 3D bone marrow cavity ROIs offers physicians and researchers a more granular view of the cell proliferation patterns, with less time invested.

The vertebral bodies are marrow compartment ROIs with high cellular proliferation activity during a patient's hematopoietic recovery to engraftment, and therefore they are a common target for analysis – both clinically and in HSCT research [7, 24]. In this thesis I have developed and presented an effective method of performing instance segmentation of individual vertebral bodies from under-sampled CT/FLT-PET image volumes of HSCT patients on the 28th day post-transplant. The image volumes have been under-sampled in the axial direction to limit the radiation exposure to patients during their vulnerable recovery towards engraftment. The under-sampled imaging protocol does not resolve the individual vertebral boundaries in the CT modality, making instance segmentation challenging. I use a semantic segmentation convolutional neural network [12] as a steppingstone to the eventual instance segmentation. Inspired by the high performance of fully-3D U-Net-derived networks on 3D medical image segmentation tasks [18, 19], but unable to implement such a model due to the high computational complexity and large memory footprint, I attempted a so-called "pseudo-3D" U-Net model to incorporate information from multiple 2D views of the object classes and thereby improve the prediction result [89, 90, 91].

Using new vertebral body, pelvis, and sternum class masks [17] for an HSCT patient dataset first introduced by Williams et. al for their FLT-PET imaging pilot study [6], I constructed multi-class, multi-view ensemble U-Net models from three constituent 2D U-

Nets. These constituent U-Nets were trained on image-mask pairs sliced from 3D patient volumes along the axial, sagittal, and coronal planes using a weighted cross entropy loss with down-weighted background class. I tested an ensemble-by-averaging strategy, following the work of Shigeta et al. [91], but also tested an ensemble-by-voting strategy and an additive ensemble method. Of the three constituent U-Net models, the axial-trained model performs the best on the test data with a mean Dice score on the vertebral body class of 0.922, with the sagittal model close behind at 0.902. The coronal model performed poorly relative to the others with a Dice score of 0.849. The only ensemble method to outperform the constituent axial-trained model on the test set was the additive ensemble method consisting of only the axial- and sagittal-trained U-Net models, which resulted in a Dice score increase of +0.0023 over the singular axial model on the vertebral body class.

The slight performance benefit is likely not worth the extra time and effort in developing additional models to use in such an ensemble configuration. I am not able to directly compare this result to the previous automatic segmentation on the HSCT patient dataset by Nguyen et al. [10, 11], since they only report true positive rate which is not a representative performance metric for the segmentation task (the true positive rate of the axial U-Net model I trained ranges anywhere from 0.0 to 1.0 depending on the selection of the classification threshold). Still, the quantitative performance of the 2D U-Net on the vertebral body semantic segmentation task is good. The qualitative performance on the pelvis and sternum classes also shows the versatility of the U-Net model. This flexibility – being able to train the same deep architecture to handle multiple object classes – combined with state-of-the-art object segmentation performance, makes convnets like the U-Net great models for image segmentation when sufficient labelled training data is available. It is very likely that the segmentation accuracy could be improved further towards state-of-the-art by using a fully-3D convnet such as 3D U-Net [18] or V-Net [19], and I would certainly recommend this for practitioners that have the computational capability.

I also presented a module for purely asymmetric super-resolution that “drops in” to most modern state-of-the-art SR convnets such as SRGAN [56], EDSR [102], and RCAN [35]. This drop-in method is possible because these SR convnets use a common architectural theme of feature extraction in the low-resolution space via long chains of convolutional residual blocks, followed by a sub-pixel [100] upsampling layer. The module I implement is a simple design that replaces the sub-pixel up-sampling used by the symmetric SR convnet with a transposed convolution layer. The transposed convolution layer has a higher computational cost, but much easier implementation for the asymmetric case, with no SR performance impact [115]. The PyTorch implementation of the transposed convolutional layer [53] implicitly allows for asymmetric upsampling by allowing asymmetric stride and asymmetric kernels, which I use to create an SR convnet based on the SRResNet generator as implemented in SRGAN [54]. To my knowledge, this is the first convnet built specifically to perform purely asymmetric super-resolution. The work from [21] provides a convnet for general asymmetric SR but they do not test the model on the *purely* asymmetric case where one dimension is held to unity while the other is scaled.

Motivated by the use-case of using SR to upscale under-sampled medical images, I created a custom task-specific medical imaging dataset for SR training. This custom dataset was built from the publicly available VerSe [29] CT volumes, from which I took high-resolution sagittal CT images containing the spine and down-sampled them along the axial direction by various scaling factors to create the low-resolution training pairs. In addition to the custom medical image dataset, I validated the SR convnet in both asymmetric and symmetric scaling modes using the DIV2K natural image dataset [116], which has been widely used in recent super-resolution convnet research. I also adapted DIV2K by making it grayscale, to match the single-channel nature of most medical imaging modalities. It was found that the symmetric models performed nearly identically on the 3-channel DIV2K dataset and the single-channel DIV2K_GRAY dataset. The asymmetric performance on DIV2K_GRAY is good, with the asymmetric 4x* model having similar reconstruction performance to the symmetric 3x model.

The VerSe-trained models performed the best on the VerSe test set, outperforming the DIV2K-trained models and the VerSe-finetuned models pre-trained on DIV2K. The asymmetric VerSe-trained models on the custom VerSe test set achieved PSNR scores of 41.72 dB for the asymmetric 2x* model, 38.83 dB for the asymmetric 3x* model, and 36.93 dB for the asymmetric 4x* model. Quantitatively, all the SR models I trained far exceeded reconstruction performance of the naïve bicubic interpolation-based method. The overall performance was limited by use of a shallower, narrower, and therefore less-capable network than the current state-of-the-art SR convnets. Baseline performance could be increased simply by using this asymmetric upsampling model on larger networks with greater representation power, such as EDSR [102] or RCAN [35]. Still, the reconstruction results are very good.

Qualitatively, the asymmetric models perform very well on the VerSe test set. I show in Figure 23 that the SR models reconstruct the complex texture of the spine with good accuracy, which is made particularly evident with the 4x* scaling factor examples. However, qualitative performance on sagittal slices from the HSCT patient dataset described in Chapter 3 is not as good as the qualitative performance seen on the VerSe test set. Appendix B shows some examples where the asymmetric SR model fails to consistently resolve the boundaries between the thoracic vertebrae. In most cases the lumbar vertebral boundaries are well resolved, but at least some thoracic vertebrae incorrectly appear fused. The lower qualitative performance on the unlearned HSCT patient dataset may be explained by one or more factors. The bicubic downsampling procedure I used in creating the low-resolution images for the VerSe training set may not be a reasonable approximation of the in-situ under-sampling of a CT volume. It is possible that training SR convnet models with low-resolution images generated from a more representative downsampling procedure may increase the performance on the HSCT patient dataset (nearest-neighbor interpolation may be a reasonable choice). Of course, there are also differences in the CT imaging protocols used for capturing the CT image volumes for each of the datasets, and these technical differences may play a role in the qualitative performance discrepancy.

The asymmetric super-resolution convnet I created proves to be a good method to accurately upscale slices from under-sampled medical image volume data. An obvious next step for this SR use case is a generative SR convnet architecture for 3D input-output training patches with the asymmetric implementation in mind. Pham et al. present a fully-3D SR convnet (albeit for symmetric upscaling) in [117], but this is a very shallow network built as a 3D extension of SRCNN [98] and lacks the representative capability of the more recent and much deeper 2D SR convnets. The main challenge with the 3D SR approach is that making a much deeper network will be computationally very expensive. EDSR is already very large at 43 million parameters, and it is just a 2D image convnet. Another area to improve is the medical imaging training set. The custom VerSe training set that I built is not particularly diverse, and I believe generalization performance of the SR convnets on the medical image upscaling task could be improved by expanding and curating high-resolution medical images to be used as a general benchmark, much like DIV2K has become the standard for training on natural image data. Lastly, the sub-pixel convolutional layer would be a more efficient alternative to the transposed convolution used in my purely asymmetric upsampling module. An implementation of asymmetric sub-pixel upsampling would be a useful tool to add to the mainstream deep learning frameworks.

Since the asymmetric SR convnet was unable to consistently resolve the vertebral boundaries in the CT modality of the HSCT patient dataset, I complete the vertebral body instance segmentation task using other methods. I present a simple algorithm that is able to robustly detect vertebral boundaries from the under-sampled post-HSCT day-28 FLT-PET image volumes. Using the semantic segmentation prediction from the U-Net model, the FLT-PET data is initially masked by the vertebral body class. Next, the dimensionality of the FLT-PET image data is reduced. This is accomplished by averaging the masked FLT-PET values in each axial slice, creating a 1D signal representing the average intensity per unit masked area. The algorithm iteratively detects valleys in this signal by using a window with a dynamic size determined by the expected vertebral span in the axial direction. This prior is adjusted (decreased) at the T9 vertebrae and T12 vertebrae to account for the relative size of vertebrae. The vertebral span prior and the adjustments were determined by empirical analysis of the FLT-PET volumes in the HSCT patient dataset. While the algorithm is simple, it robustly detects vertebral boundaries from the 28th-day FLT-PET image volumes that are undetectable in the CT modality.

I also showed a method for visualization of segmented FLT-PET image data. The method uses isosurfaces of FLT-PET intensity levels to create a "heatmap-like" 3D view of cell proliferation within a given bone structure ROI. FLT-PET isosurface values are selected by constucting discrete cumulative distribution functions from histograms of the FLT-PET ROIs. I found setting isosurfaces at the approximate 50th, 85th, and 98th percentiles of the discrete FLT-PET cumulative distribution creates a good visualization of cell proliferation activity. Many of these visualizations are shown in Appendix D, and the method is generalizable to work with any combination of mask and PET data. Lastly, I calculated SUV content of the segmented ROIs by applying the segmentation masks for the pelvis, sternum, and individual vertebral bodies to the FLT-PET image volumes.

In sum, I have developed a practical method for the instance segmentation of individual vertebral bodies from under-sampled 28th day joint CT/FLT-PET image volume data. This automatic instance segmentation of 3D bone marrow compartment ROIs in the FLT-PET modality provides a more granular view of the cell proliferation patterns in HSCT patients, with less time invested compared to conventional "by-hand" drawing of ROIs. The method starts with a semantic segmentation U-Net model to create a prediction mask of the vertebral body bone column. The vertebral body column is used to mask the FLT-PET volume data, and the masked FLT-PET data is then used to locate the boundaries between the vertebrae. Finally, the vertebral body mask is sliced at the detected boundaries to reach the final instance segmentation. The algorithm is very robust, with only a single mis-identified vertebral boundary in the available HSCT patient data.

6.1 Original Contributions

The specific original contributions I made in this thesis include:

- A multi-view 2D ensemble multi-class U-Net model for the simultaneous semantic segmentation of the column of vertebral bodies, the pelvis, and the sternum from CT image data. For the HSCT patient dataset, the flexibility of the U-Net model to predict multiple complex unconnected bone structures is an improvement over the previous methods, which were hand-crafted to segment only the vertebral bodies [10].
- An asymmetric upsampling module based on transposed convolutional layers that can be easily “dropped in” to many existing state-of-the-art single-image SR convnets to enable purely asymmetric SR. Asymmetric SR convnets may be useful as a preprocessing step in place of naïve interpolation methods to reconstruct under-sampled medical images with anisotropic voxels to higher-resolution isotropic volumes. The asymmetric SR task was trained and tested on both the DIV2K natural image dataset [116] and a custom medical image dataset sampled from the VerSe dataset [28].
- A boundary detection algorithm that enforces a vertebra size prior to detect the boundaries between the individual vertebrae in under-sampled CT/FLT-PET volumes. When combined with the vertebral body column segmentation of Chapter 3 this allows for the instance segmentation of vertebral bodies, even in cases where a human cannot distinguish the vertebral boundaries in the CT modality. The method I implemented produces lower RMSE (for the detected boundaries) than the Kalman filter method shown by [10], in some cases by large margins. However, it must be noted that in [10] the authors include the cervical vertebrae, which I omit.
- A tool for automatically generating 3D visualizations based on isosurfaces of the FLT-PET image volume data in segmented ROIs. The visualizations may help medical researchers analyze the FLT-PET data in greater detail.

6.2 Recommendations for Future Research

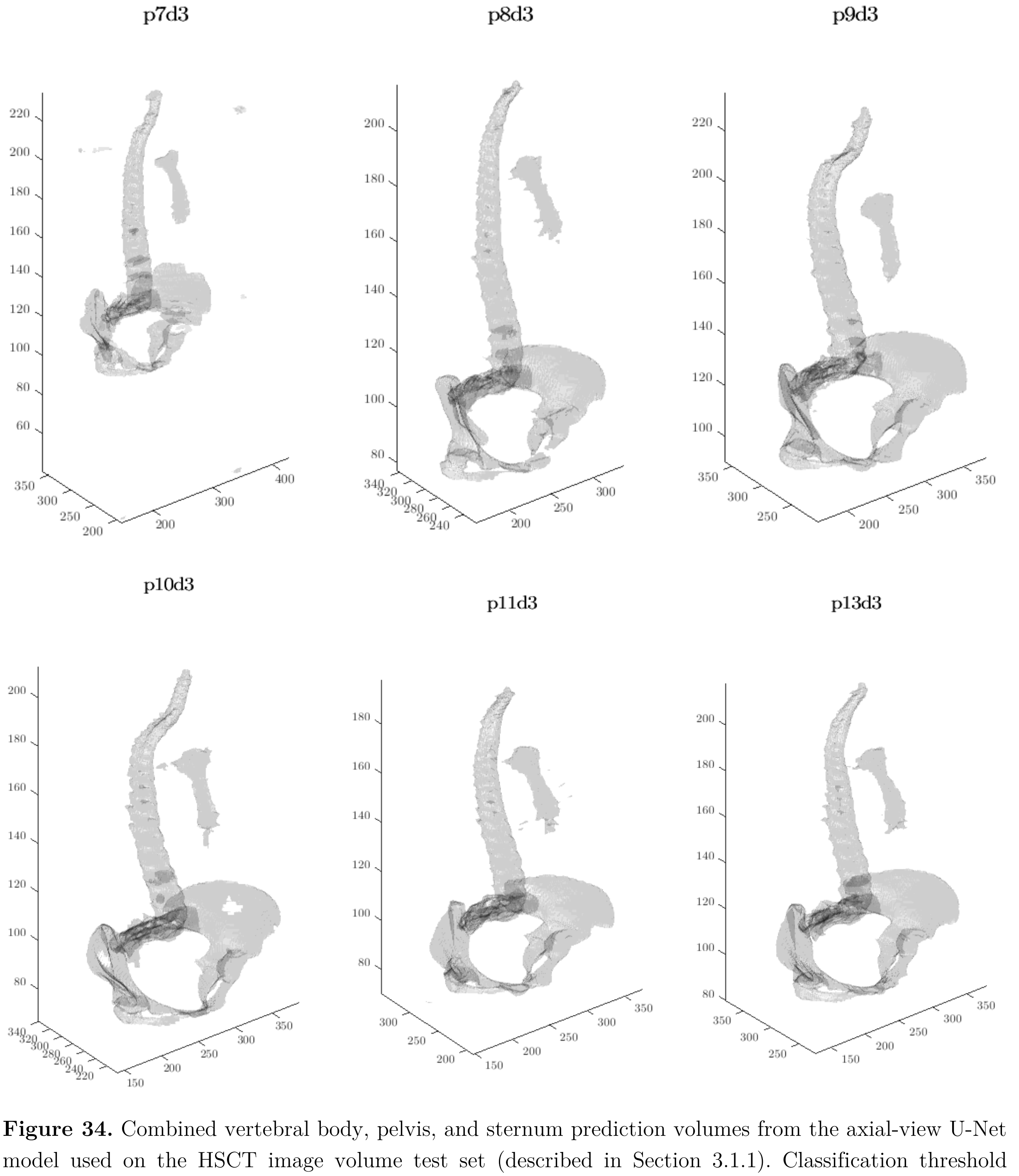
Based on my work in this thesis, I have targeted the following areas for potential future research:

- Implementation of a 3D U-Net [18] (or similar 3D autoencoder image segmentation architecture [19]) may increase the segmentation performance for the multiclass segmentation task that is pursued in Chapter 3. In the Large-Scale Vertebrae Segmentation Challenge (VerSe) [28], the winning model was a 3D U-Net that achieved a segmentation Dice Score of 0.917. As I noted in Section 2.3.4, dataset inconsistencies make it hard to make definitive comparisons between models trained and tested on different datasets. Testing a fully-3D U-Net-like model on the existing patient data would allow a direct comparison of the 2D and 3D methods on the HSCT patient dataset.
- Increasing the number of object segmentation classes to include more bone marrow cavity regions of interest (such as the bones of the arm and leg) could be a great help towards further studying the proliferation of hematopoietic stem cells in marrow cavities throughout the body in the days and weeks post-HSCT. This manual labelling would be a time-consuming task, but the data may be useful enough to warrant the investment. Additionally, a more comprehensive and diverse CT volume dataset with many types of labelled bone structures could be helpful even for applications outside of the HSCT use case that is discussed in this thesis.
- While the reconstruction results for the asymmetric super-resolution task of Chapter 4 are very good (particularly when compared to naïve methods like bicubic interpolation), the overall performance was limited by use of a shallower, narrower, and therefore less-capable network than the current state-of-the-art SR convnets. Baseline performance could be increased simply by using this asymmetric upsampling model on larger networks with greater representation power, such as EDSR [102] or RCAN [35].
- As mentioned in Section 4.3, there is a qualitative performance discrepancy between the HSCT patient dataset and the VerSe dataset on the asymmetric SR task. Performance appears to be better on the test data from the VerSe dataset, particularly at the intentioned sub-task of resolving the vertebral boundaries. The bicubic downsampling procedure I used in creating the low-resolution images for the VerSe training set may not be a reasonable approximation of the in-situ undersampling of a CT volume. It is possible that training SR convnet models with low-resolution images generated from a more representative downsampling procedure

may increase the performance on the HSCT patient dataset. A variety of downsampling procedures could be tested to see if a more representative method exists.

- Generalization performance of the SR convnets on the general medical image upscaling task could be improved by expanding and curating high-resolution medical images to be used as a training dataset and general benchmark, much like DIV2K has become the standard for training on natural image data. This could be done for various imaging modalities such as X-ray, CT, MR, and PET. In contrast to the manual labelling of ground-truth image segmentation data, super-resolution training data is relatively easy to create, requiring only high-resolution ground truth examples from which low-resolution examples can be algorithmically derived.
- The sub-pixel convolutional layer [100] would be a more efficient alternative to the transposed convolution used in my purely asymmetric upsampling module. An implementation of asymmetric sub-pixel upsampling would be a useful tool to add to the mainstream deep learning frameworks.
- The vertebral boundary detection algorithm of Chapter 5 may be improved by allowing segmentation of the cervical vertebral bodies in addition to the lumbar and thoracic. One idea to increase accuracy in the cervical spine region is to adjust the vertebral prior upon every new detection of a vertebral boundary, instead of at the defined values of $d_{\mathrm{T12}} = \lceil \frac{4}{5} d_{\mathrm{L4}} \rceil$, $d_{\mathrm{T7}} = \lceil \frac{3}{4} d_{\mathrm{L4}} \rceil$, and $d_{\mathrm{T4}} = \lceil \frac{3}{5} d_{\mathrm{L4}} \rceil$ which were determined though analysis of the ground-truth data.

Appendix A.
Multi-Class Prediction Volumes



p7d3

p8d3

p9d3

p10d3

p11d3

p13d3

Figure 34. Combined vertebral body, pelvis, and sternum prediction volumes from the axial-view U-Net model used on the HSCT image volume test set (described in Section 3.1.1). Classification threshold is set to $p = 0.5$. “pNd3” indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

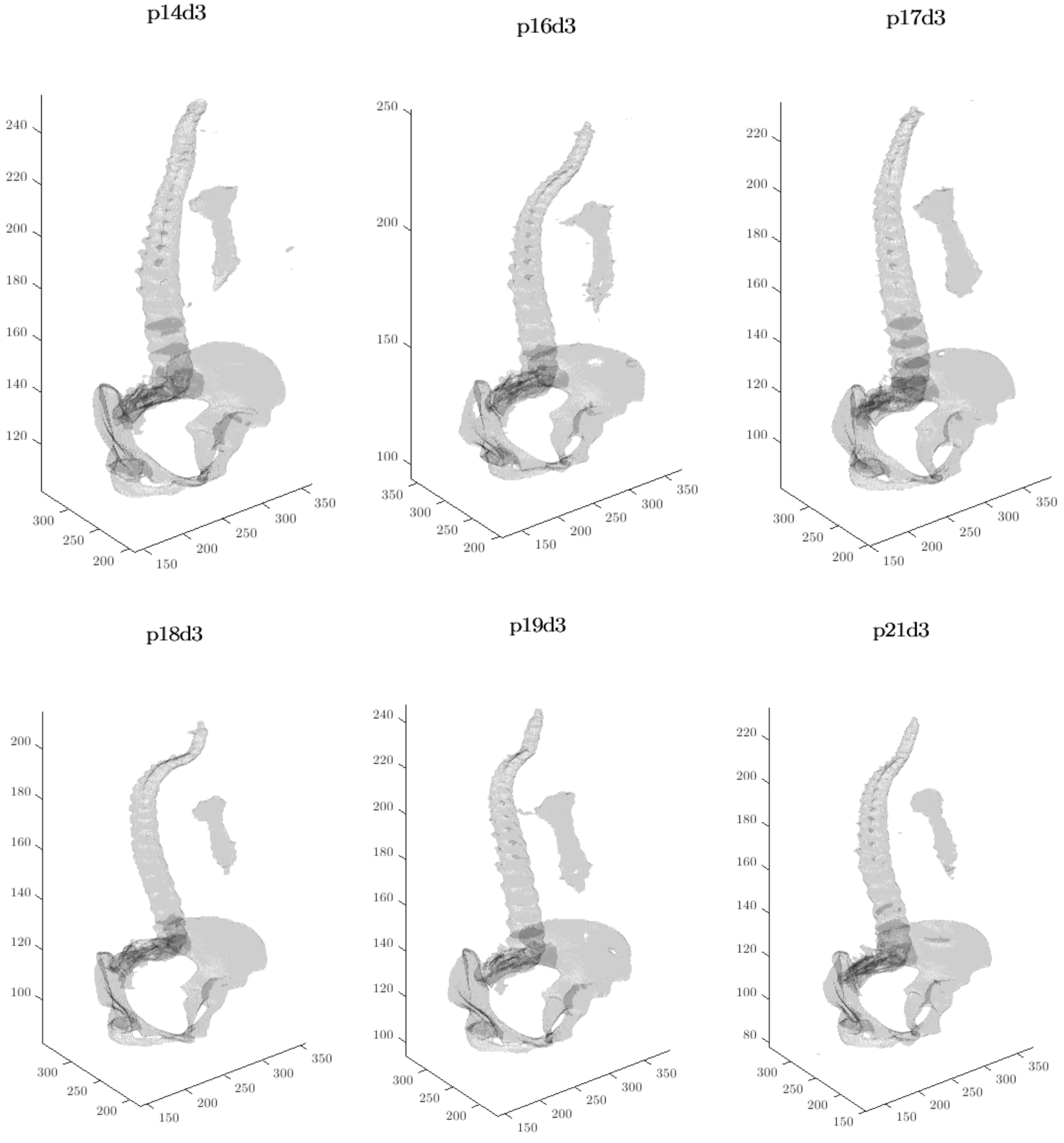


Figure 35. Combined vertebral body, pelvis, and sternum prediction volumes from the axial-view U-Net model used on the HSCT image volume test set (described in Section 3.1.1). Classification threshold is set to $p = 0.5$. "pNd3" indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where "d3" is approximately the 28th day post-HSCT).

Appendix B.

Asymmetric SR on the HSCT Patient Dataset

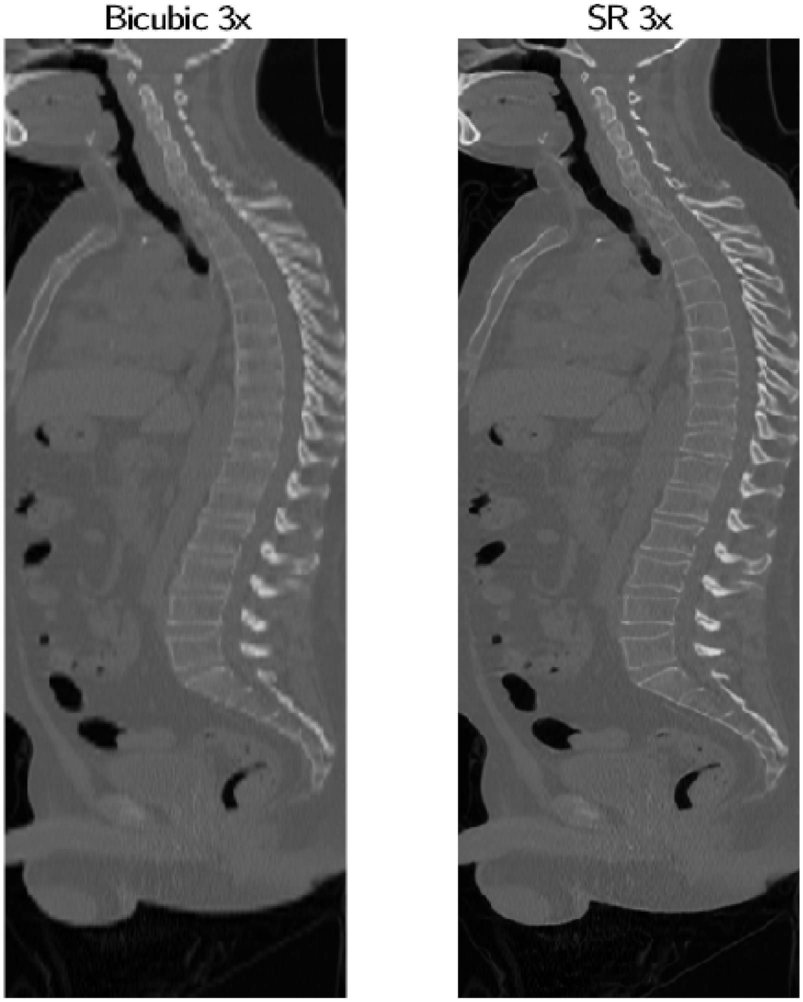


Figure 36. Comparison between the bicubic upsampling method and the convnet-based asymmetric SR upsampling model introduced in Chapter 4. The low-resolution image from which this example is produced is sampled from the HSCT dataset introduced in Chapter 3.

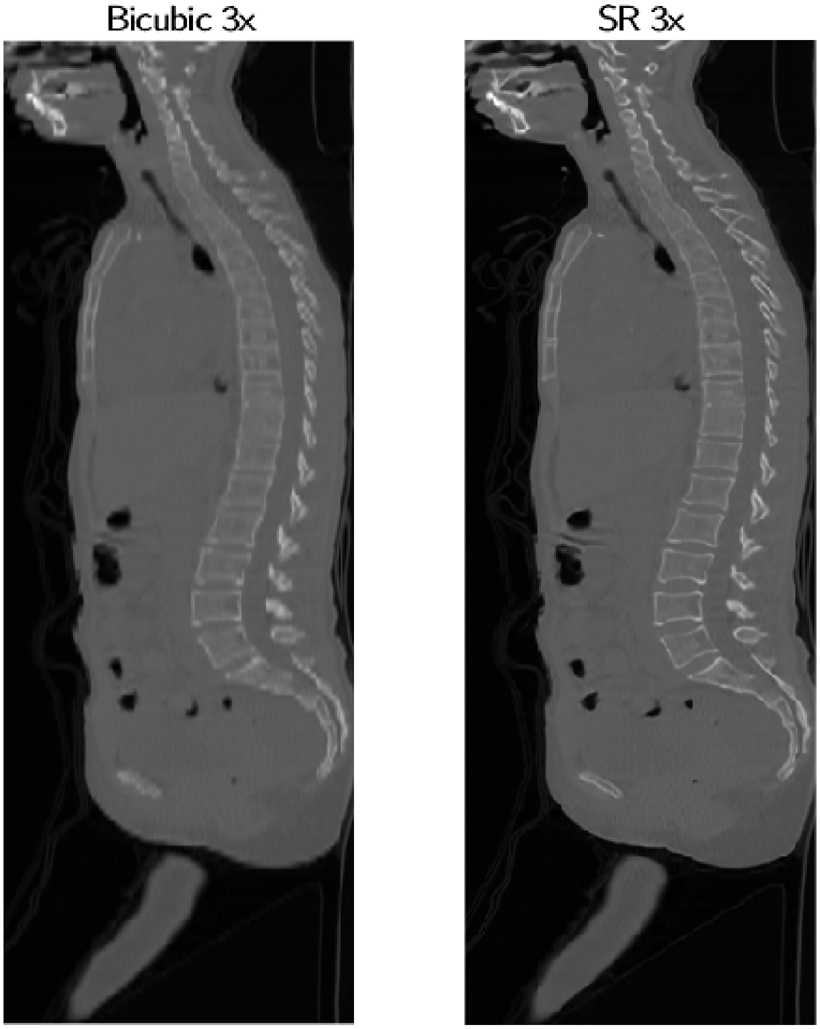


Figure 37. Comparison between the bicubic upsampling method and the convnet-based asymmetric SR upsampling model introduced in Chapter 4. The low-resolution image from which this example is produced is sampled from the HSCT dataset introduced in Chapter 3.

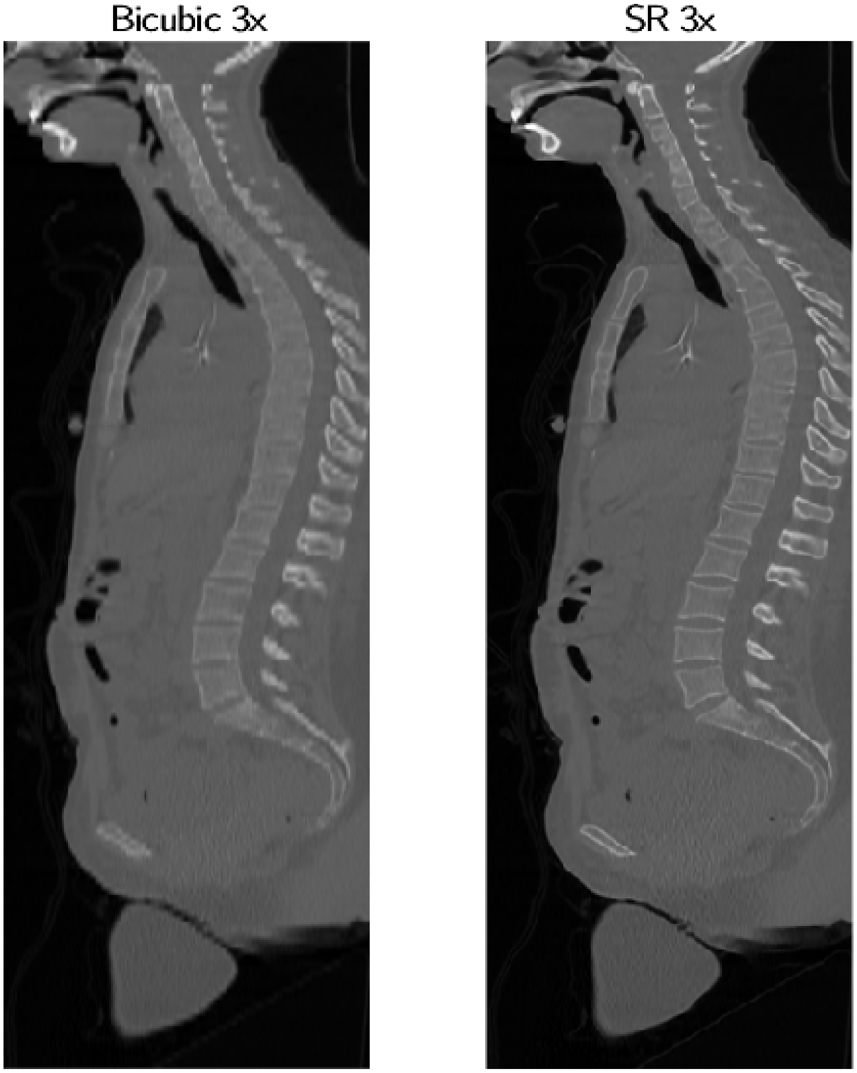


Figure 38. Comparison between the bicubic upsampling method and the convnet-based asymmetric SR upsampling model introduced in Chapter 4. The low-resolution image from which this example is produced is sampled from the HSCT dataset introduced in Chapter 3.

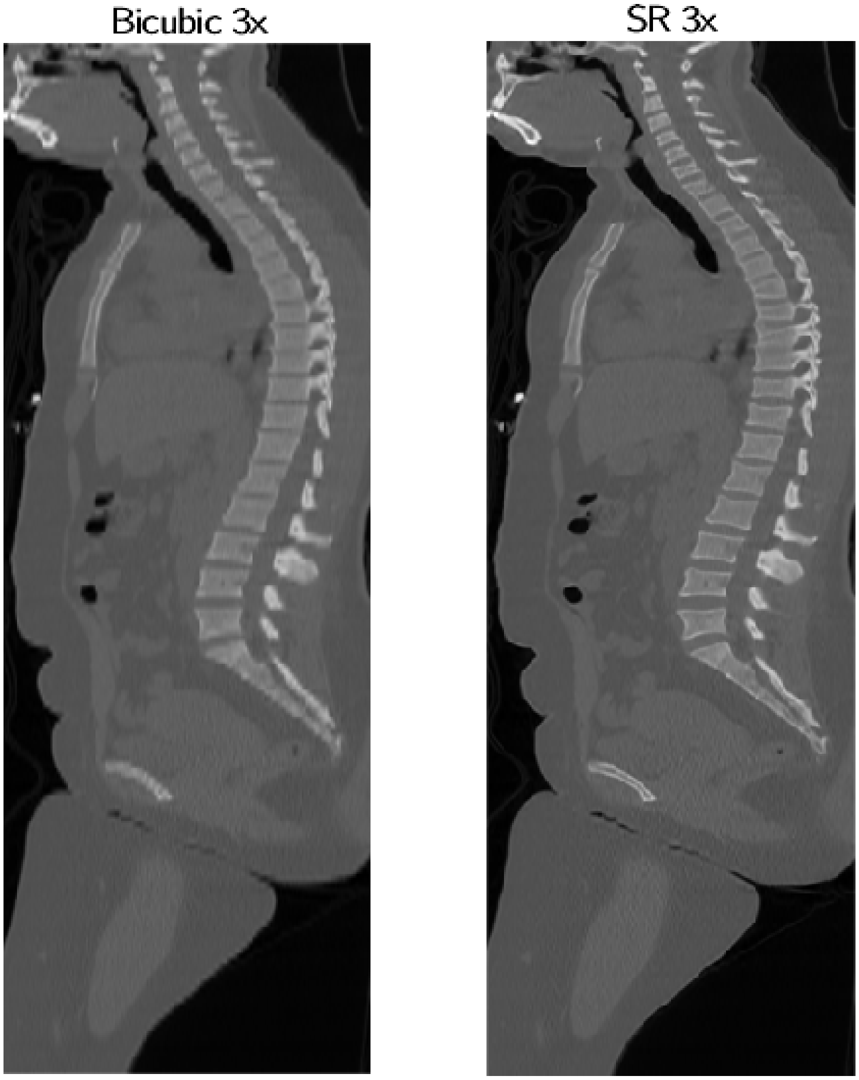


Figure 39. Comparison between the bicubic upsampling method and the convnet-based asymmetric SR upsampling model introduced in Chapter 4. The low-resolution image from which this example is produced is sampled from the HSCT dataset introduced in Chapter 3.

Appendix C.

Instance Segmentation of Vertebral Bodies

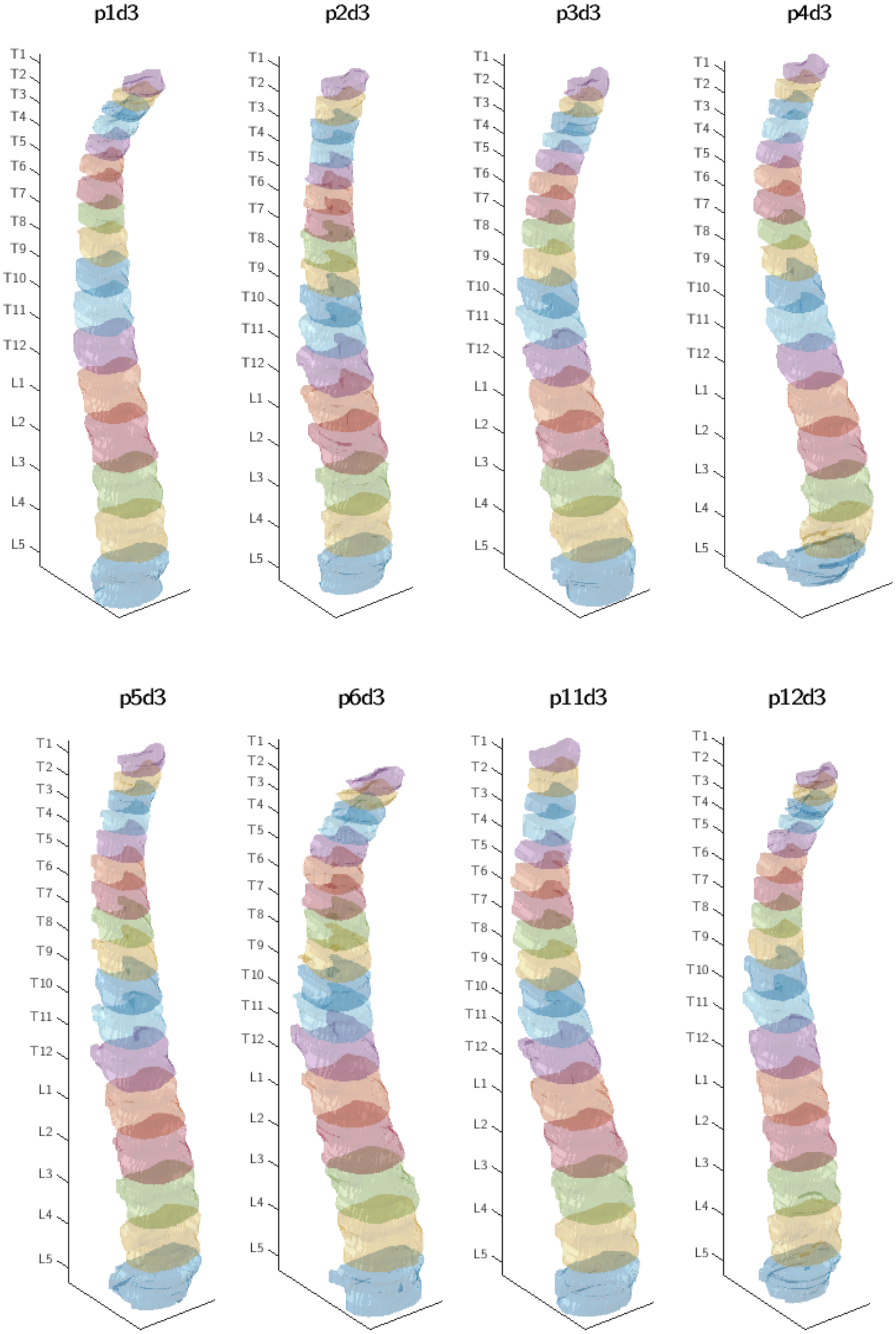


Figure 40. Instance segmentation of vertebral bodies by the method shown in Chapter 5. “pNd3” indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

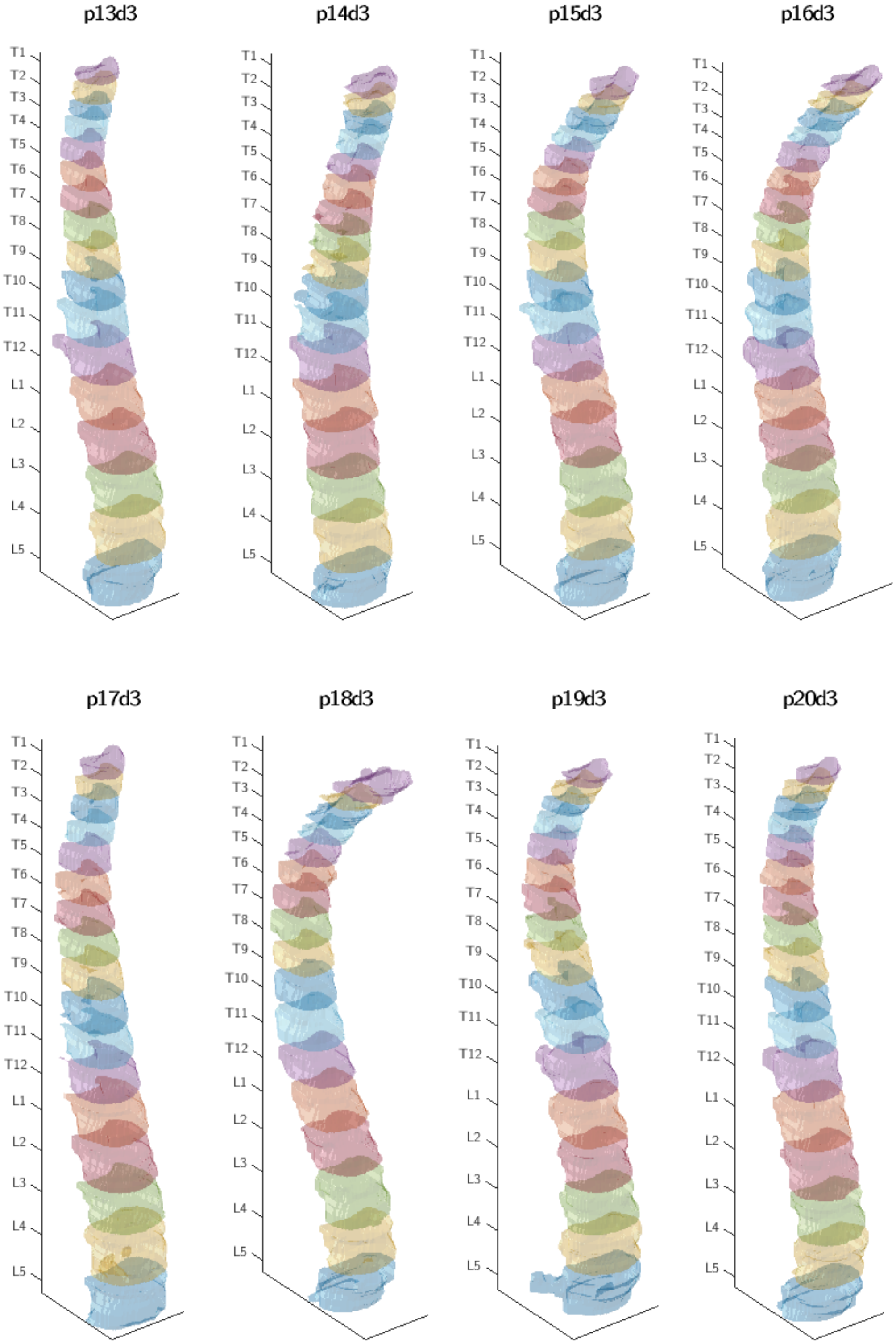


Figure 41. Instance segmentation of vertebral bodies by the method shown in Chapter 5. “pNd3” indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

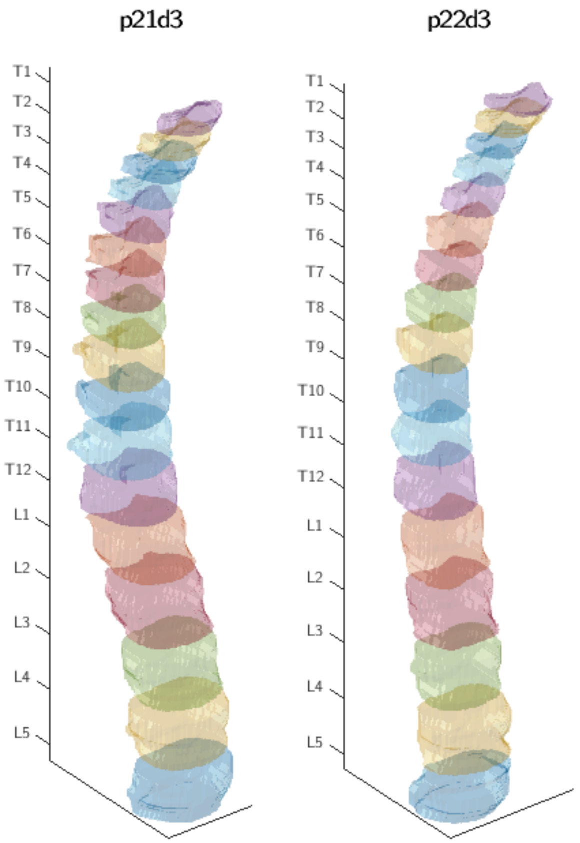


Figure 42. Instance segmentation of vertebral bodies by the method shown in Chapter 5. “pNd3” indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

Appendix D.

FLT-PET Visualization in Bone Marrow ROIs

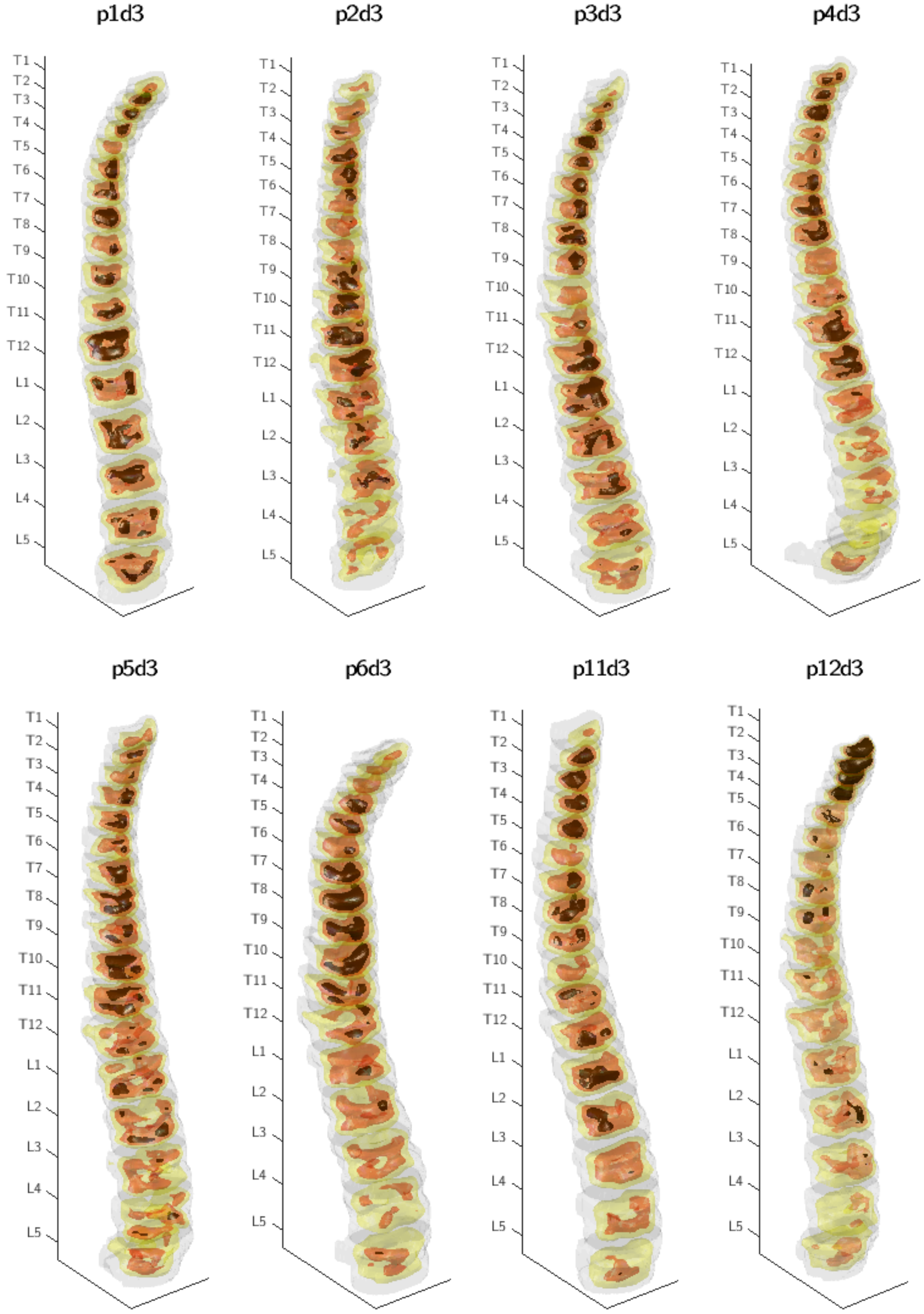


Figure 43. FLT-PET visualization via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. "pNd3" indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where "d3" is approximately the 28th day post-HSCT).

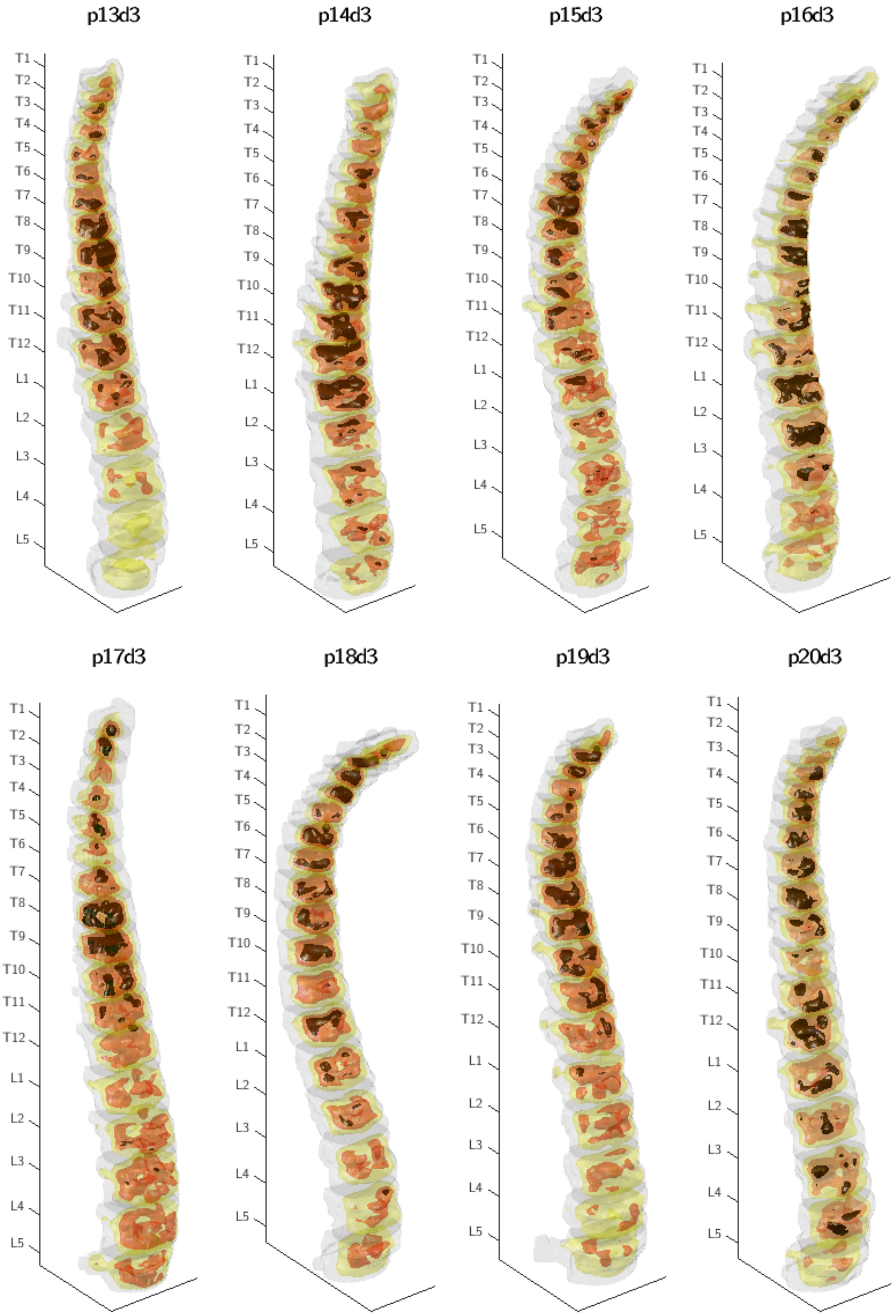


Figure 44. FLT-PET visualization via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. “pNd3” indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

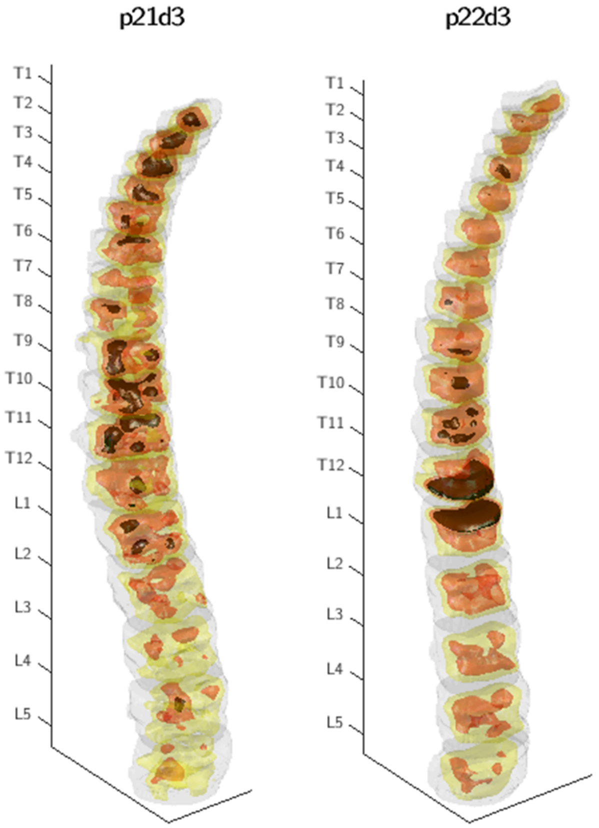


Figure 45. FLT-PET visualization via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. “pNd3” indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

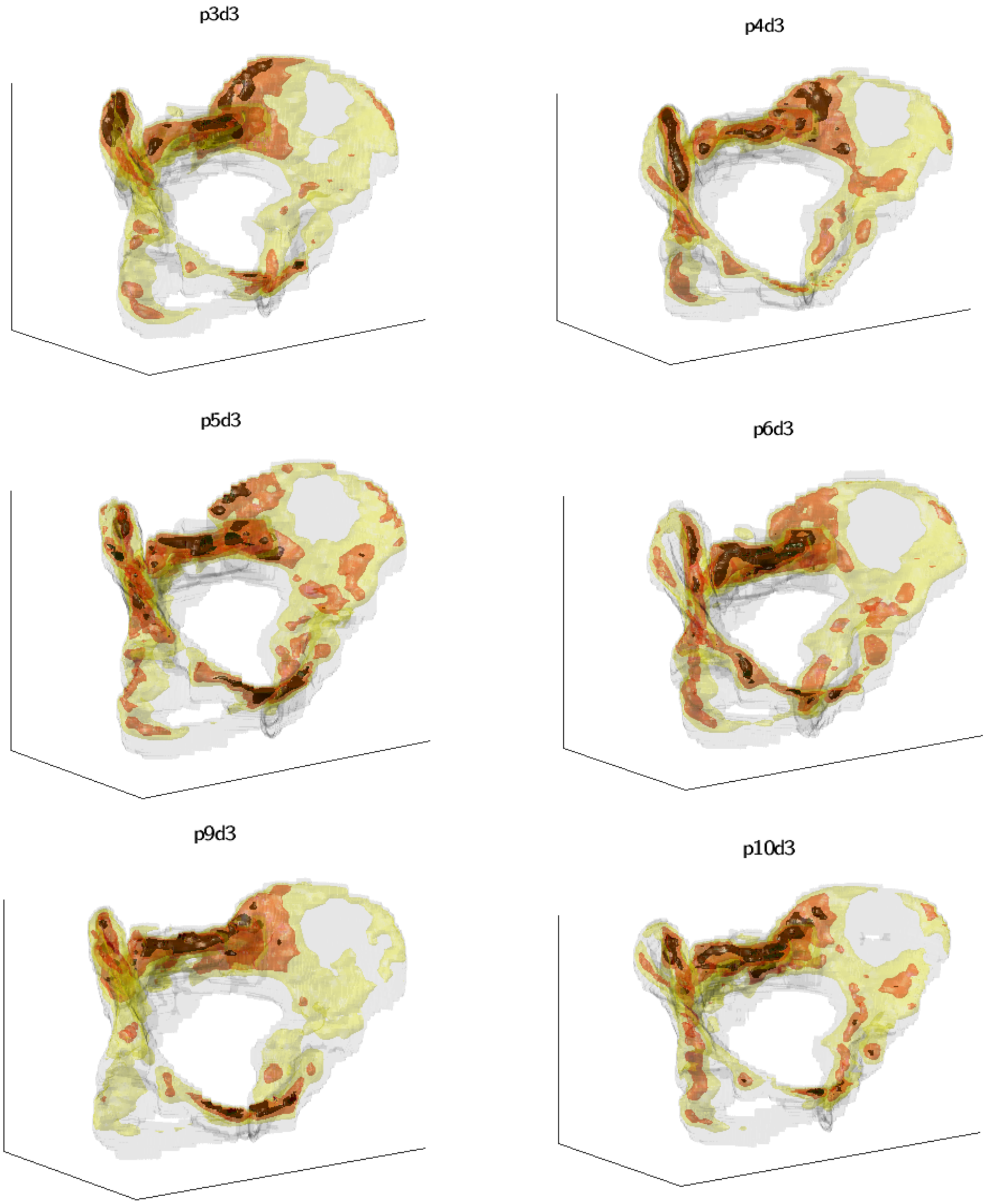


Figure 46. FLT-PET visualization in the pelvis ROI via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. “pXd3” indicates the patient number and scan index in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

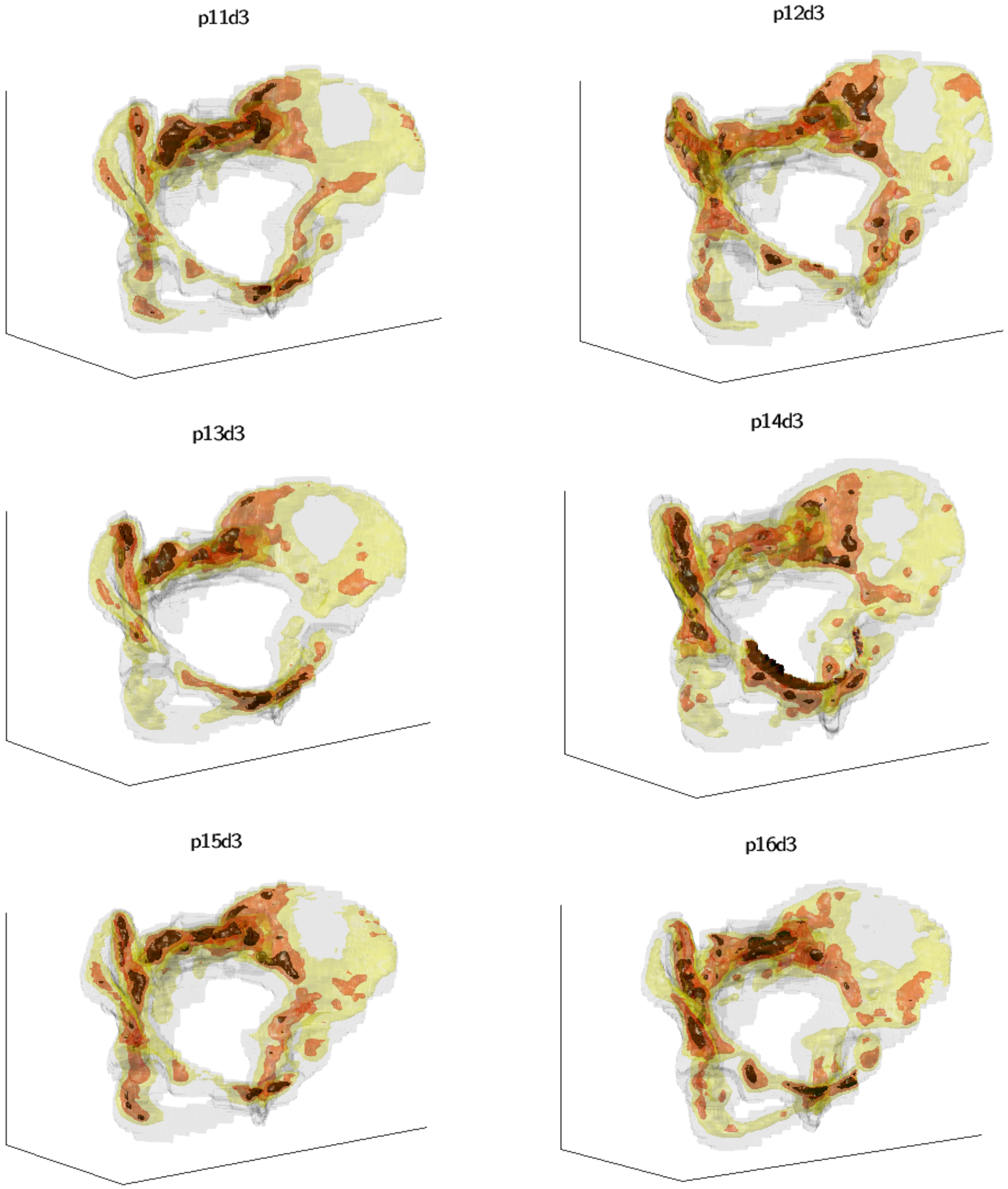


Figure 47. FLT-PET visualization in the pelvis ROI via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. “pXd3” indicates the patient number and scan index in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

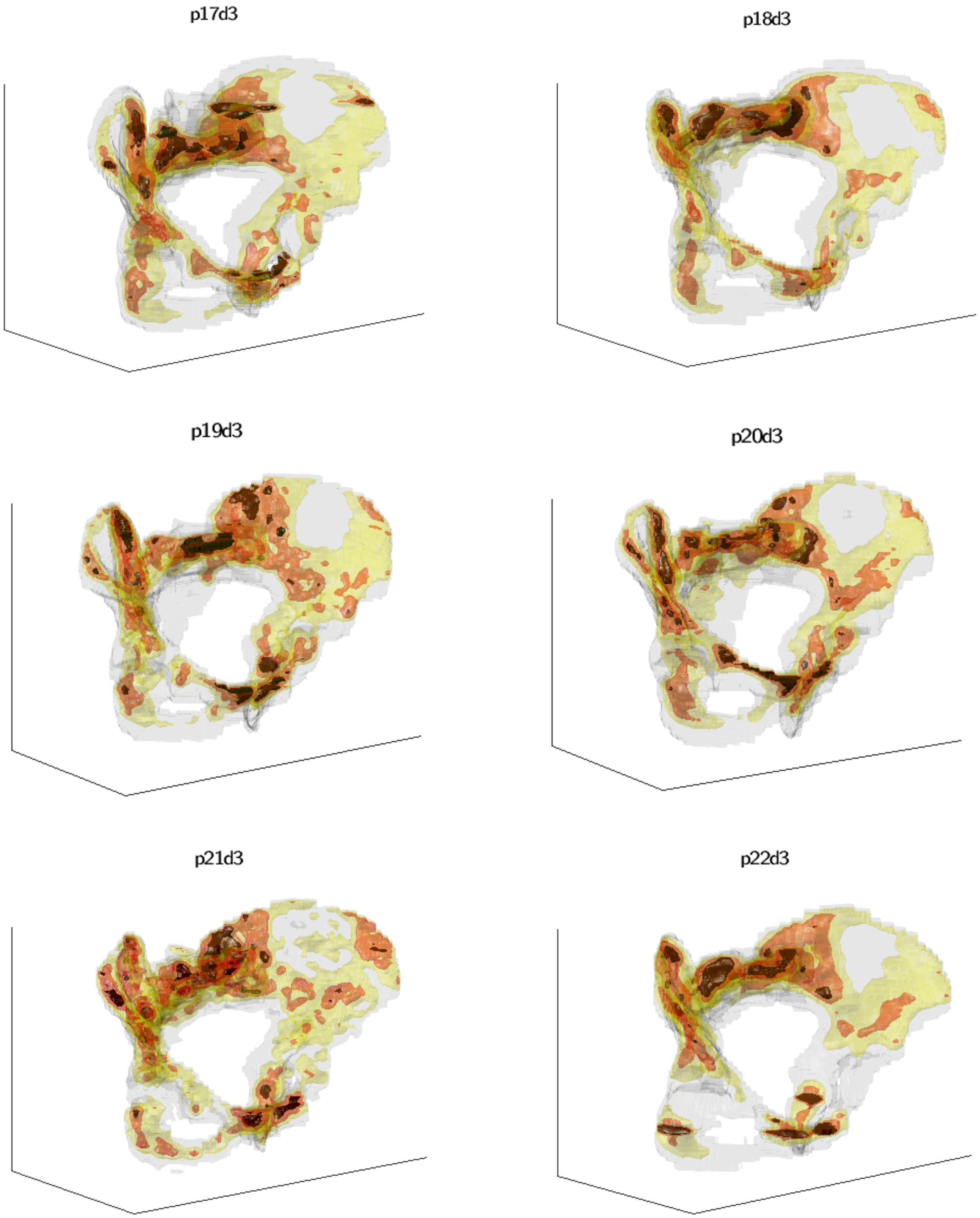


Figure 48. FLT-PET visualization in the pelvis ROI via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. “pXd3” indicates the patient number and scan index in the anonymized HSCT dataset (where “d3” is approximately the 28th day post-HSCT).

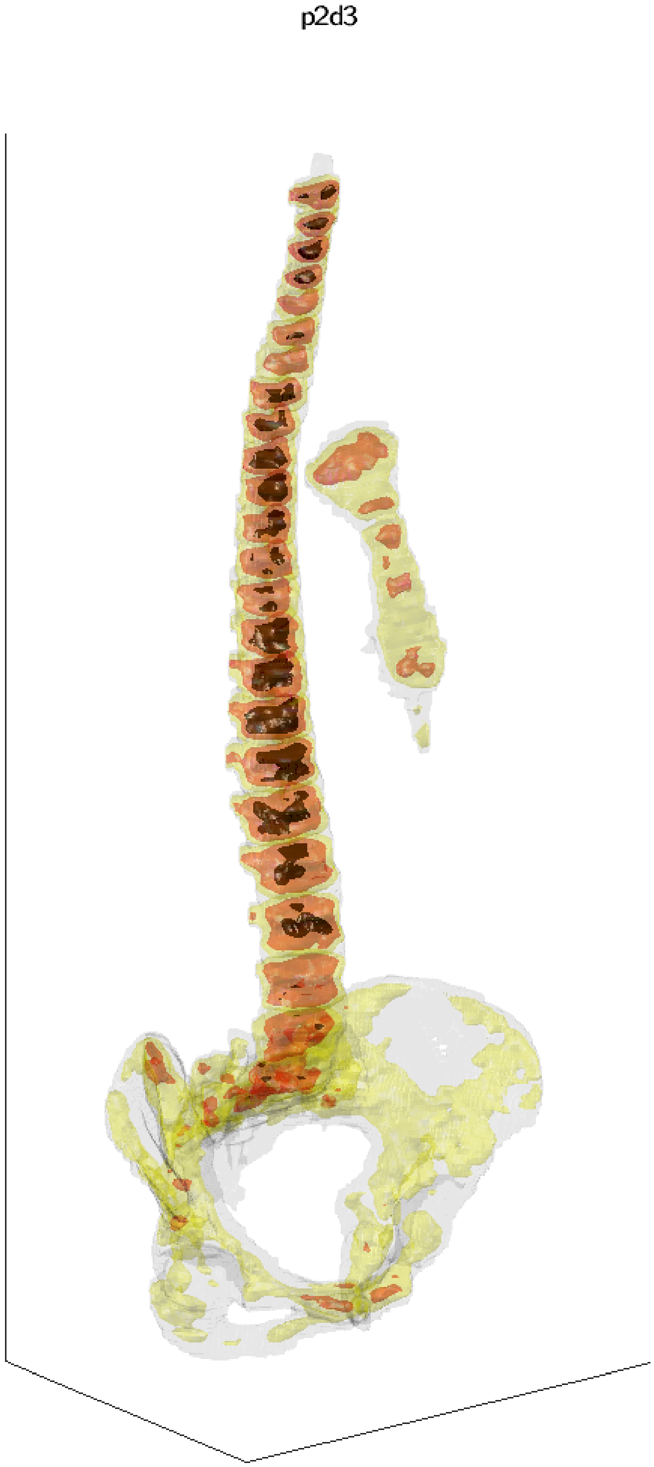


Figure 49. FLT-PET visualization of the combined vertebral body, pelvis, and sternum ROIs via isosurfaces. Yellow surfaces represent 50th percentile FLT-PET values within the masked volume, orange represents the 85th percentile, red represents the 98th percentile. "pNd3" indicates the patient number N and scan index d3 in the anonymized HSCT dataset (where "d3" is approximately the 28th day post-HSCT).

Appendix E.
SUV Extraction Tables

TABLE 7. SUV FOR VERTEBRAL BODY ROIS ON THE 28TH DAY POST-HSCT.

VB	p22	p21	p20	p19	p18	p17	p16	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1
L5	4295	1881	4544	5005	3485	3707	4604	4407	4074	5979	2842	6265	4852	3978	1736	3465	6057	5369	5191	4347	2534	5555
L4	4928	2079	5276	4817	3275	4345	4760	4027	4820	5747	2846	6091	5088	3645	1750	3894	6370	5360	4810	3966	2857	5531
L3	4720	2117	4765	4861	3344	4799	4549	4278	4899	5724	3217	5851	5167	3820	1954	4456	6299	5585	5149	3765	3041	5321
L2	4625	2250	4762	4699	2912	4111	4266	3792	4606	5545	3163	5727	4315	3507	1835	4678	6083	5587	5218	3347	2989	5111
L1	5159	2235	4694	4638	3093	3828	4010	3647	4928	5662	2876	5723	4425	3561	1802	4642	6072	5617	4700	3353	2874	4802
T12	4938	2146	4383	5031	2761	3527	4153	3812	4928	5681	2747	4860	3977	3248	1739	3771	5234	4756	5285	2820	2597	4377
T11	3507	2069	3494	3995	2860	3166	3141	3467	4165	5266	2173	3757	4268	3012	1273	3484	4350	4101	4767	2190	2130	3444
T10	3466	1647	2515	3731	2425	2923	2711	2947	3623	3937	1925	3128	3217	2545	1354	3328	4033	3464	3570	1649	1996	2738
T9	3017	1775	2867	3596	1959	2856	2201	2579	2708	3590	1689	3190	2723	2521	973	2789	3592	2987	3017	1592	1826	2361
T8	2636	1261	2541	2978	2169	2445	1980	2502	2361	3281	1427	3322	2534	2034	893	2364	3469	2602	2622	1520	1257	2040
T7	1926	1140	2252	2487	1665	1703	1585	2279	2306	2337	1239	3052	2091	1831	1157	2462	2622	2295	2359	1163	1191	1931
T6	2067	1204	1946	2079	1521	1198	1181	1799	1967	1989	1095	2566	1561	1412	767	2068	2522	1861	2380	1021	1152	1518
T5	1532	1092	1383	1459	1209	1268	1357	1429	1595	1681	987	1907	1624	1393	823	1594	1849	1762	1720	927	889	1278
T4	1427	814	1300	1264	914	1090	857	1319	1312	1588	657	2265	1009	1095	652	1628	1668	1396	1640	661	968	1247
T3	1031	833	1314	1292	880	819	1178	1010	1491	1251	867	1752	1244	572	437	1336	1213	1220	1253	822	785	933
T2	1439	725	929	1407	922	819	1515	1335	1329	1633	998	1972	911	1158	638	1172	1137	1267	1525	642	925	1051
T1	790	529	549	1205	573	956	793	1032	1110	2055	626	1309	914	936	416	928	967	950	1236	677	674	839

SUV extraction method is outlined in Chapter 5.

Table 8. SUV for VBC*, Sternum, and Pelvis ROIs Across 3 Imaging Days

PELVIS	**p22**	**p21**	**p20**	**p19**	**p18**	**p17**	**p16**	**p15**	**p14**	**p13**	**p12**	**p11**	**p10**	**p9**	**p8**	**p7**	**p6**	**p5**	**p4**	**p3**	**p2**	**p1**
1st scan	6416	12895	9572	11544	5111	14516	14740	10371	18071	15306	9267	8929	9316	43528	12932	21693	11326	9454	7763	7835	9688	7998
2nd scan	7726	21399	-	13236	8148	7952	10371	12912	38110	14599	-	15078	16125	12526	10486	16562	19605	10106	13062	9466	13020	7828
3rd scan	70244	29333	66948	68098	55187	59468	69105	67166	74802	90640	48395	88198	67332	45127	29992	59016	87545	70750	89862	46345	33164	57956

VBC	**p22**	**p21**	**p20**	**p19**	**p18**	**p17**	**p16**	**p15**	**p14**	**p13**	**p12**	**p11**	**p10**	**p9**	**p8**	**p7**	**p6**	**p5**	**p4**	**p3**	**p2**	**p1**
1st scan	3498	8485	6523	6132	2756	8928	5881	8003	7505	5106	3302	5027	4522	7865	7838	9149	7802	7425	4540	4131	6833	5859
2nd scan	5558	31614	-	9505	4979	6655	8003	5102	40584	6044	-	10319	10160	8465	7180	12724	15287	9438	7995	6060	10412	7873
3rd scan	61382	30630	63095	63968	42875	49412	52517	54042	61472	72727	35564	75834	57536	47140	24882	56730	74643	64991	66604	39660	36763	56654

STERNU	**p22**	**p21**	**p20**	**p19**	**p18**	**p17**	**p16**	**p15**	**p14**	**p13**	**p12**	**p11**	**p10**	**p9**	**p8**	**p7**	**p6**	**p5**	**p4**	**p3**	**p2**	**p1**
1st scan	571	1080	1164	1478	408	1511	1028	1208	1098	706	460	796	709	440	745	767	1803	953	553	636	925	797
2nd scan	787	2152	-	1622	727	1339	1208	801	2082	769	-	1206	1509	971	699	1142	3008	998	890	689	1144	934
3rd scan	4930	2565	4640	4957	3805	3738	3933	2078	4961	7241	3243	8916	5049	3517	2114	6175	13221	6655	6090	3447	3923	5745

VBC is "Vertebral Body Column" – the semantic segmentation.

1st scan is day before HSCT (ablated); 2nd scan is between 5 and 9 days post-HSCT; 3rd scan is 28th day post-HSCT.

SUV extraction method is outlined in Chapter 5.

References

[1] Z.-Q. Zhao, P. Zheng, S.-t. Xu and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 30, pp. 3212–3232, 2019.

[2] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1-22, 2021.

[3] J. T. Bushberg and J. M. Boone, The Essential Physics of Medical Imaging, Lippincott Williams & Wilkins, 2011.

[4] N. C. Dalrymple, S. R. Prasad, F. M. El-Merhi and K. N. Chintapalli, "Price of isotropy in multidetector CT," *Radiographics,* vol. 27, pp. 49–62, 2007.

[5] S. Schelhaas, K. Heinzmann, V. R. Bollineni, G. M. Kramer, Y. Liu, J. C. Waterton, E. O. Aboagye, A. F. Shields, D. Soloviev and A. H. Jacobs, "Preclinical Applications of 3'-Deoxy-3'-[(18)F]Fluorothymidine in Oncology - A Systematic Review," *Theranostics,* vol. 7, pp. 40–50, 1 2017.

[6] K. M. Williams, J. Holter-Chakrabarty, L. Lindenberg, Q. Duong, S. K. Vesely, C. T. Nguyen, J. P. Havlicek, K. Kurdziel, J. Gea-Banacloche, F. I. Lin, D. N. Avila, G. Selby, C. G. Kanakry, S. Li, T. Scordino, S. Adler, C. M. Bollard, P. Choyke and R. E. Gress, "Imaging of subclinical haemopoiesis after stem-cell transplantation in patients with haematological malignancies: a prospective pilot study," *The Lancet Haematology,* vol. 5, pp. e44–e52, 2018.

[7] K. M. Williams and J. H. Chakrabarty, "Imaging haemopoietic stem cells and microenvironment dynamics through transplantation," *The Lancet Haematology,* vol. 7, pp. e259–e269, 2020.

[8] A. Agool, R. H. J. A. Slart, P. M. Kluin, J. T. M. de Wolf, R. A. J. O. Dierckx and E. Vellenga, "F-18 FLT PET: a noninvasive diagnostic tool for visualization of the bone marrow compartment in patients with aplastic anemia: a pilot study," *Clinical Nuclear Medicine,* vol. 36, pp. 286–289, 2011.

[9] A. F. Shields, J. R. Grierson, B. M. Dohmen, H.-J. Machulla, J. C. Stayanoff, J. M. Lawhorn-Crews, J. E. Obradovich, O. Muzik and T. J. Mangner, "Imaging Proliferation In Vivo with [F-18] FLT and Positron Emission Tomography," *Nature medicine,* vol. 4, pp. 1334–1336, 1998.

[10] C. Nguyen, J. Havlicek, Q. Duong, S. Vesely, R. Gress, L. Lindenberg, P. Choyke, J. H. Chakrabarty and K. Williams, "An automatic 3D CT/PET segmentation framework for bone marrow proliferation assessment," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016.

[11] C. T. Nguyen, J. P. Havlicek, J. H. Chakrabarty, Q. Duong and S. K. Vesely, "Towards automatic 3D bone marrow segmentation," in *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, 2016, pp. 4126-4130.

[12] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234-241.

[13] A. Klein, J. Warszawski, J. Hillengaß and K. H. Maier-Hein, "Towards whole-body CT bone segmentation," in *Bildverarbeitung für die Medizin 2018*, Springer, 2018, pp. 204–209.

[14] H. Dong, G. Yang, F. Liu, Y. Mo and Y. Guo, "Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks," in *Annual Conference on Medical Image Understanding and Analysis (MIUA)*, 2017, pp. 506-517.

[15] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 2018, p. 3–11.

[16] N. Kumar, R. Verma, D. Anand, Y. Zhou, O. F. Onder, E. Tsougenis, H. Chen, P.-A. Heng, J. Li, Z. Hu and others, "A multi-organ nucleus segmentation challenge," *IEEE transactions on medical imaging,* vol. 39, pp. 1380–1391, 2019.

[17] C. T. Nguyen, *personal communication,* 2020.

[18] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2016, pp. 424-432.

[19] F. Milletari, N. Navab and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *International Conference on 3D Vision (3DV)*, 2016, pp. 565-571.

[20] Z. Wang, J. Chen and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22 pp., 2020.

[21] L. Wang, Y. Wang, Z. Lin, J. Yang, W. An and Y. Guo, "Learning for Scale-Arbitrary Super-Resolution from Scale-Specific Networks," *arXiv preprint arXiv:2004.03791,* 2020.

[22] M. Jagannathan-Bogdan and L. I. Zon, "Hematopoiesis," *Development,* vol. 140, pp. 2463–2467, 2013.

[23] S. J. Morrison and J. Kimble, "Asymmetric and Symmetric Stem-Cell Divisions in Development and Cancer," *Nature,* vol. 441, pp. 1068–1074, 2006.

[24] R. R. Jenq and M. R. M. Van den Brink, "Allogeneic haematopoietic stem cell transplantation: individualized stem cell and immune therapy of cancer," *Nature Reviews. Cancer,* vol. 10, pp. 309–309, 2010.

[25] D. Hutt, "Engraftment, graft failure, and rejection," *The European Blood and Marrow Transplantation Textbook for Nurses,* pp. 259–270, 2018.

[26] K. Mah and C. B. Caldwell, "Biological target volume," *PET-CT in Radiotherapy Treatment Planning,* pp. 52–89, 2008.

[27] R. Boellaard, N. C. Krak, O. S. Hoekstra and A. A. Lammertsma, "Effects of noise, image resolution, and ROI definition on the accuracy of standard uptake values: a simulation study," *Journal of Nuclear Medicine,* vol. 45, pp. 1519–1527, 2004.

[28] A. Sekuboyina, M. E. Husseini, A. Bayat, M. Löffler, H. Liebl, H. Li, G. Tetteh, J. Kukačka, C. Payer, D. Štern, M. Urschler, M. Chen, D. Cheng, N. Lessmann, Y. Hu, T. Wang, D. Yang, D. Xu, F. Ambellan, T. Amiranashvili, M. Ehlke, H. Lamecker, S. Lehnert, M. Lirio, N. P. de Olaguer, H. Ramm, M. Sahu, A. Tack, S. Zachow, T. Jiang, X. Ma, C. Angerman, X. Wang, K. Brown, M. Wolf, A. Kirszenberg, É. Puybareau, D. Chen, Y. Bai, B. H. Rapazzo, T. Yeah, A. Zhang, S. Xu, F. Hou, Z. He, C. Zeng, Z. Xiangshang, X. Liming, T. J. Netherton, R. P. Mumme, L. E. Court, Z. Huang, C. He, L.-W. Wang, S. H. Ling, L. D. Huynh, N. Boutry, R. Jakubicek, J. Chmelik, S. Mulay, M. Sivaprakasam, J. C. Paetzold, S. Shit, I. Ezhov, B. Wiestler, B. Glocker, A. Valentinitsch, M. Rempfler, B. H. Menze and J. S. Kirschke, "VerSe: A Vertebrae Labelling and Segmentation Benchmark for Multi-detector CT Images"*, arXiv preprint arXiv:2001.09193,* 2021.

[29] M. T. Löffler, A. Sekuboyina, A. Jacob, A.-L. Grau, A. Scharr, M. El Husseini, M. Kallweit, C. Zimmer, T. Baum and J. S. Kirschke, "A vertebral segmentation dataset with fracture grading," *Radiology: Artificial Intelligence,* vol. 2, p. e190138, 2020.

[30] A. Sekuboyina, M. Rempfler, A. Valentinitsch, B. H. Menze and J. S. Kirschke, "Labeling Vertebrae with Two-dimensional Reformations of Multidetector CT Images: An Adversarial Approach for Incorporating Prior Knowledge of Spine Anatomy," *Radiology: Artificial Intelligence,* vol. 2, p. e190074, 3 2020.

[31] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision,* vol. 128, pp. 261–318, 2020.

[32] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[33] H. Gray and W. H. Lewis, Anatomy of the human body, Philadelphia: Lea & Febiger, 1918.

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

[35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286-301.

[36] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review,* vol. 53, pp. 5455–5516, 2020.

[37] D. Scherer, A. Müller and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *International conference on artificial neural networks*, 2010, pp. 92-101.

[38] B. Xu, N. Wang, T. Chen and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853,* 2015.

[39] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *ICML*, 2010, pp. 807-814.

[40] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv preprint arXiv:1702.05659,* 2017.

[41] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging,* vol. 3, pp. 47–57, 2016.

[42] J. Duchi, E. Hazan and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research,* vol. 12, pp. 2121-2159, 2011.

[43] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701,* 2012.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101,* 2017.

[46] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747,* 2016.

[47] S. Boyd, S. P. Boyd and L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[48] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983,* 2016.

[49] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*, 2017, pp. 464-472.

[50] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019, p. 1100612.

[51] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics,* vol. 4, pp. 1–17, 1964.

[52] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, pp. 533–536, 1986.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy,

B. Steiner, L. Fang, J. Bai and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems (NeurIPS) 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[54] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,* 2015, Available: https://www.tensorflow.org/.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448-456.

[56] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang and others, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4681-4690.

[57] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data,* vol. 6, pp. 1–48, 2019.

[58] S. C. Wong, A. Gatt, V. Stamatescu and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?," in *2016 International conference on digital image computing: techniques and applications (DICTA)*, pp. 6, 2016.

[59] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," vol. 2, 2007.

[60] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *International journal of multimedia information retrieval,* pp. 1–19, 2020.

[61] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu and R. Urtasun, "Polytransform: Deep polygon transformer for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9131-9140.

[62] N. Salman, "Image segmentation based on watershed and edge detection techniques.," *International Arab Journal of Information Technology,* vol. 3, pp. 104–110, 2006.

[63] F. Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," *Image and Vision Computing,* vol. 23, pp. 877–886, 2005.

[64] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE transactions on pattern analysis and machine intelligence,* vol. 32, pp. 604–618, 2010.

[65] Z. Huang, X. Wang, J. Wang, W. Liu and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014-7023.

[66] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 679–698, 1986.

[67] J. Zhang, C.-H. Yan, C.-K. Chui and S.-H. Ong, "Fast segmentation of bone in CT images using 3D adaptive thresholding," *Computers in Biology and Medicine,* vol. 40, pp. 231–236, 2010.

[68] N. Dhanachandra, K. Manglem and Y. J. Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science,* vol. 54, pp. 764–771, 2015.

[69] M.-Y. Liu, O. Tuzel, S. Ramalingam and R. Chellappa, "Entropy rate superpixel segmentation," in *CVPR 2011*, 2011, pp. 2097-2104.

[70] F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *2012 International conference on systems and informatics (ICSAI2012)*, 2012, pp. 1936-1941.

[71] S. Korman, D. Reichman, G. Tsur and S. Avidan, "Fast-match: Fast affine template matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2331-2338.

[72] A. M. Christensen, N. V. Passalacqua and E. J. Bartelink, "Chapter 12 - Individual Skeletal Variation," in *Forensic Anthropology*, A. M. Christensen, N. V. Passalacqua and E. J. Bartelink, Eds., San Diego, Academic Press, 2014, pp. 301-339.

[73] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3D medical image segmentation: a review," *Medical Image Analysis,* vol. 13, pp. 543–563, 2009.

[74] A. Neubert, J. Fripp, C. Engstrom, R. Schwarz, L. Lauer, O. Salvado and S. Crozier, "Automated detection, 3D segmentation and analysis of high resolution spine MR images using statistical shape models," *Physics in Medicine & Biology,* vol. 57, pp. 8357, 2012.

[75] D. Forsberg, "Atlas-based segmentation of the thoracic and lumbar vertebrae," in *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, Springer, 2015, pp. 215–220.

[76] J. Yao, J. E. Burns, D. Forsberg, A. Seitel, A. Rasoulian, P. Abolmaesumi, K. Hammernik, M. Urschler, B. Ibragimov, R. Korez and others, "A multi-center milestone study of clinical vertebral CT segmentation," *Computerized Medical Imaging and Graphics,* vol. 49, pp. 16–28, 2016.

[77] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters,* vol. 27, pp. 861–874, 2006.

[78] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging,* vol. 15, pp. 1–28, 2015.

[79] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857,* 2017.

[80] V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, pp. 2481–2495, 2017.

[81] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence,* vol. 40, pp. 834–848, 2017.

[82] D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 15, pp. 850–863, 1993.

[83] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis,* vol. 42, pp. 60–88, 2017.

[84] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431-3440.

[85] R. Augustaukas and A. Lipnickas, "Pixel-wise Road Pavement Defects Detection Using U-Net Deep Neural Network," in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2019, pp. 468-471.

[86] Z. Zhang, Q. Liu and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters,* vol. 15, pp. 749–753, 2018.

[87] X. Yang, X. Li, Y. Ye, X. Zhang, H. Zhang, X. Huang and B. Zhang, "Road Detection via Deep Residual Dense U-Net," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, 7 pp.

[88] A. G. Smith, J. Petersen, R. Selvan and C. R. Rasmussen, "Segmentation of roots in soil with u-net," *Plant Methods,* vol. 16, pp. 1–15, 2020.

[89] Q. Yu, Y. Xia, L. Xie, E. K. Fishman and A. L. Yuille, "Thickened 2D networks for efficient 3D medical image segmentation," *arXiv preprint arXiv:1904.01150,* 2019.

[90] C. Angermann and M. Haltmeier, "Random 2.5 D U-net for Fully 3D Segmentation," in *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting*, Springer, 2019, pp. 158–166.

[91] N. Shigeta, M. Kamata and M. Kikuchi, "Effectiveness of Pseudo 3D Feature Learning for Spinal Segmentation by CNN with U-Net Architecture," *Journal of Image and Graphics,* vol. 7, pp. 107–111, 2019.

[92] A. Arbelle and T. R. Raviv, "Microscopy cell segmentation via convolutional LSTM networks," in *International Symposium on Biomedical Imaging (ISBI)*, 2019, pp. 1008-1012.

[93] N. Lessmann, B. Van Ginneken, P. A. De Jong and I. Išgum, "Iterative fully convolutional neural networks for automatic vertebra segmentation and identification," *Medical Image Analysis,* vol. 53, pp. 142–155, 2019.

[94] J. Yang, J. Wright, T. S. Huang and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing,* vol. 19, pp. 2861–2873, 2010.

[95] R. Timofte, V. De Smet and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision (ACCV)*, 2014, pp. 111-126.

[96] R. Timofte, V. De Smet and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1920-1927.

[97] H. Chang, D.-Y. Yeung and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, 8 pp.

[98] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 38, pp. 295–307, 2015.

[99] K. He, X. Zhang, S. Ren and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, 2016, pp. 630-645.

[100] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874-1883.

[101] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[102] B. Lim, S. Son, H. Kim, S. Nah and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017, pp. 4681-4690.

[103] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2472-2481.

[104] S. Anwar and N. Barnes, "Densely Residual Laplacian Super-Resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 12 pp., 2020.

[105] T. Dai, J. Cai, Y. Zhang, S.-T. Xia and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11065-11074.

[106] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan and J. Sun, "Meta-SR: A magnification-arbitrary network for super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1575-1584.

[107] S. A. Taghanaki, K. Abhishek, J. P. Cohen, J. Cohen-Adad and G. Hamarneh, "Deep semantic segmentation of natural and medical images: a review," *Artificial Intelligence Review,* vol. 54, pp. 137–178, 2021.

[108] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu and G. Hamarneh, "Combo loss: Handling input and output imbalance in multi-organ segmentation," *Computerized Medical Imaging and Graphics,* vol. 75, pp. 24–33, 2019.

[109] milesial, *Pytorch-UNet,* 2020, Available: https://github.com/milesial/Pytorch-UNet.

[110] M. Martinez and R. Stiefelhagen, "Taming the cross entropy loss," in *German Conference on Pattern Recognition*, 2018, pp. 628-637.

[111] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980-2988.

[112] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 2017, pp. 240–248.

[113] M. Berman, A. R. Triki and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4413-4421.

[114] T. H. Phan and K. Yamamoto, "Resolving class imbalance in object detection with weighted cross entropy losses," *arXiv preprint arXiv:2006.01413,* 2020.

[115] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig and Z. Wang, "Is the deconvolution layer the same as a convolutional layer?," *arXiv preprint arXiv:1609.07009,* 2016.

[116] R. Timofte, E. Agustsson, L. V. Gool, M.-H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, X. Wang, Y. Tian, K. Yu, Y. Zhang, S. Wu, C. Dong, L. Lin, Y. Qiao, C. C. Loy, W. Bae, J. Yoo, Y. Han, J. C. Ye, J.-S. Choi, M. Kim, Y. Fan, J. Yu, W. Han, D. Liu, H. Yu, Z. Wang, H. Shi, X. Wang, T. S. Huang, Y. Chen, K. Zhang, W. Zuo, Z. Tang, L. Luo, S. Li, M. Fu, L. Cao, W. Heng, G. Bui, T. Le, Y. Duan, D. Tao, R. Wang, X. Lin, J. Pang, J. Xu, Y. Zhao, X. Xu, J. Pan, D. Sun, Y. Zhang, X. Song, Y. Dai, X. Qin, X.-P. Huynh, T. Guo, H. S. Mousavi, T. H. Vu, V. Monga, C. Cruz, K. Egiazarian, V. Katkovnik, R. Mehta, A. K. Jain, A. Agarwalla, C. V. S. Praveen, R. Zhou, H. Wen, C. Zhu, Z. Xia, Z. Wang and Q. Guo, "NTIRE 2017 Challenge on

Single Image Super-Resolution: Methods and Results," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1110-1121.

[117] C.-H. Pham, A. Ducournau, R. Fablet and F. Rousseau, "Brain MRI super-resolution using deep 3D convolutional networks," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 197-200.

[118] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)," *Geoscientific Model Development Discussions,* vol. 7, pp. 1525–1534, 2014.