M.Sc. Engg. (CSE) Thesis

A Deep Learning Based Framework for 3D Reconstruction of Femur from QCT Images

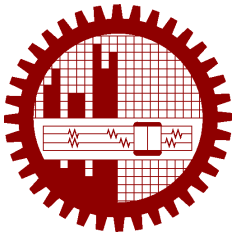
Submitted by

Jamalia Sultana

0421052011

Supervised by

Dr. Mahmuda Naznin



Submitted to

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

June 2023

Candidate's Declaration

I, do, hereby, certify that the work presented in this thesis, titled, "A Deep Learning Based Framework for 3D Reconstruction of Femur from QCT Images", is the outcome of the investigation and research carried out by me under the supervision of Dr. Mahmuda Naznin, Professor, Department of CSE, BUET.

I also declare that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Jamalia Sultana

Jamalia Sultana

0421052011

The thesis titled "**A Deep Learning Based Framework for 3D Reconstruction of Femur from QCT Images**", submitted by Jamalia Sultana, Student ID 0421052011, Session April 2021, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents on June 21, 2023.

Board of Examiners

1. Dr. Mahmuda Naznin — Chairman (Supervisor)
   Professor
   Department of CSE, BUET, Dhaka

2. Dr. Mahmuda Naznin — Member (Ex-Officio)
   Professor and Head
   Department of CSE, BUET, Dhaka

3. Dr. A.B.M. Alim Al Islam — Member
   Professor
   Department of CSE, BUET, Dhaka

4. Dr. Md. Shamsuzzoha Bayzid — Member
   Associate Professor
   Department of CSE, BUET, Dhaka

5. Dr. Swakkhar Shatabda — Member (External)
   Professor
   Department of CSE
   United International University, Dhaka

Acknowledgement

I am thankful to Dr. Mahmuda Naznin for her supervision of this work. She helped the research in shaping, deciding the steps of our work, and providing support. I would like to express my sincere thanks to Dr. Tanvir R. Faisal and his of Musculoskeletal Mechanics & Multiscale Materials (4Mlab at ULL) Lab, Department of Mechanical Engineering, University of Louisiana at Lafayette, USA, for conceiving the ideas, collaborating, all intellectual contributions, feedback, and guidance throughout the whole process of this research work.

I would like to thank Dr. Ahmed Suparna Bahar Moni, Assistant Professor, Orthopedic Surgery, University of Toledo, USA for his valuable feedback.

I would like to express my sincere gratitude to the Board of Examiners, Prof. Dr. A.B.M. Alim Al Islam and Dr. Md. Shamsuzzoha Bayzid for their valuable feedback and review. I would like to extend my heartfelt appreciation to Prof. Dr. Swakkhar Shatabda of the department of CSE at United International University for generously devoting his time to support this thesis.

I also greatly acknowledge Sarah Doll of Mechanical Engineering at University of Louisiana at Lafayette, USA and Mashiyat Nayeem of Computer Science and Engineering of North South University, Bangladesh for helping in data annotation.

This is an excellent blending of Computer Science and Engineering and Musculoskeletal Mechanics & Multiscale Material Science. The research journey has been excellent.

Dhaka
June 21, 2023

Jamalia Sultana
0421052011

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Deep Learning (DL) based techniques have been proven to be very effective in medical image segmentation and reconstruction of 3D anatomies of a human body. The success of DL methods primarily depends on extensive and accurately annotated datasets. In this research, we propose a semi-supervised deep learning method that we call SSDL utilizing a CNN-based 3D U-Net model for femur segmentation from sparsely annotated Quantitative Computed Tomography (QCT) slices. We have focused only on annotating QCT slices at the proximal end of the femur, forming ball and socket joint with acetabulum for precise segmentation. Using our proposed framework, we generate a segmenting binary mask to segment the femur accurately. Using our proposed framework, a modified DICOM file integrated with the original metadata can be generated, preserving all the required information. We have employed polynomial spline interpolation for 3D reconstruction along with an Island Removal algorithm to eliminate noises, if there is any. The performance of segmentation and 3D reconstruction have been evaluated both qualitatively and quantitatively. Our approach can achieve a Dice Similarity Coefficient of 91.8% for unseen patients and 99.2% for validated patients. Additionally, we have obtained an average Relative Error of 6.61% and 12.08% for volume and surface area, respectively. The proposed approach demonstrates its effectiveness in accurately segmenting and reconstructing the 3D femur from QCT slices.

Chapter 1

Introduction

Three-dimensional (3D) reconstruction of bones from 2D image datasets has emerged as a valuable tool in orthopedics for various purposes, including preoperative planning, diagnosis, prevention, and treatment of bone fractures. The utilization of 3D bone models has been shown to significantly enhance the accuracy and reliability of orthopedic evaluations, particularly in assessing the risk of osteoporotic fractures [10]. Osteoporotic fractures, especially in the hip region, pose a significant health concern, with implications for disability and mortality, as well as substantial social and economic burdens. As the global population continues aging, the prevalence of osteoporosis and hip fractures is expected to rise. Therefore, there is a pressing need to develop strategies for early detection and prevention of hip-bone fractures in elderly individuals.

A key component in addressing this challenge is the use of 3D reconstructed femur models that enable comprehensive analyses of mechanical, morphological, and densitometric properties of femur. By examining these properties, clinicians can perform non-invasive assessments of fracture risk in a timely manner, allowing for early interventions and targeted preventive measures. Numerous studies have highlighted the importance of 3D bone reconstruction for orthopedic applications. Researchers including Ellis *et al.* [11], Otomaru *et al.* [12], and some others have demonstrated the utility of 3D reconstruction in preoperative planning and diagnosis. Meanwhile, Sheehan *et al.* [13] and Siebenlist *et al.* [14] have emphasized its significance in assessing osteoporotic fracture risk. These studies underscore the potential of 3D femur reconstruction as a valuable tool in addressing the increasing burden of osteoporotic hip fractures, allowing for early and non-invasive assessments of fracture risk in the elderly population.

The 3D reconstruction of femurs holds great promise for analyzing mechanical, morphological, and densitometric properties, leading to early detection and prevention of hip fractures, particularly in the aging population [15]. By leveraging the advantages of 3D bone models, healthcare professionals can make informed decisions and implement effective strategies to mitigate the social and economic impact of osteoporosis related fractures [16].

1.1 Motivation

Clinically, early prediction of osteoporotic hip fracture is done by measuring *Bone Mineral Density* (*BMD*) using *Dual-Energy X-ray Absorptiometry* (*DXA*) [17], that usually does not consider the 3D bone morphology. Imaging modalities such as *Quantitative Computed Tomography* (*QCT*) or *Magnetic Resonance Imaging* (*MRI*) can provide detailed 3D spatial information on the target anatomical structure [18, 19], which can be effectively used in building computational models to understand, examine, and predict the performance of individual structure of bone. To consider structural and geometric parameters along with BMD, the QCT image has been used to predict the fracture risk [20] of proximal femur coupled with *Finite Element Analysis* (*FEA*)-based computational modeling [21–25]. Traditionally, the 3D reconstruction of the proximal femur requires expensive and advanced software for biomedical image analysis, extensive well-trained user intervention, and advanced engineering and image processing knowledge that restrict the clinical use of mechanistic models as a non-invasive assessment tool till today. Therefore, we have hypothesized that a fully automated data-driven modeling framework for obtaining 3D femur can be a key and useful tool for advancing pre-clinical and clinical diagnosis and assessment of hip fracture.

1.2 Challenges

The major challenge of 3D reconstruction of the femur is the segmentation of the proximal part of femur. The femur bones connected to the pelvic bones (shown in Figure 1.1), such as acetabulum, create a complex bone geometry. Additionally, the surrounding bones exhibit a similar none density to that of the proximal femur. For this reason, it becomes challenging to distinguish the femur bone from the neighboring pelvic bones. Furthermore, notable geometric differences exist between the cancellous region of the femur and the femoral cortex.

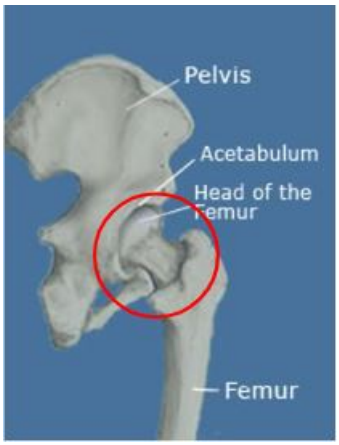


Figure 1.1: An example of complex bone geometry of femur [1].

Concurrent segmentation processes mainly involve statistical and semi-automated techniques such as primitive shape recognition methods [26], atlas-based segmentation [6, 7, 27], graph-cut approaches [8, 28], active model [9, 29], and statistical shape models [28], which usually require time consuming user monitoring and significant manual intervention (Figure 1.2). We have mentioned already that, this modeling and analysis require a high level of knowledge, expertise, and computational time. Thus, traditional biomedical software has been proven less efficient and unsuitable for health professionals for clinical usefulness.

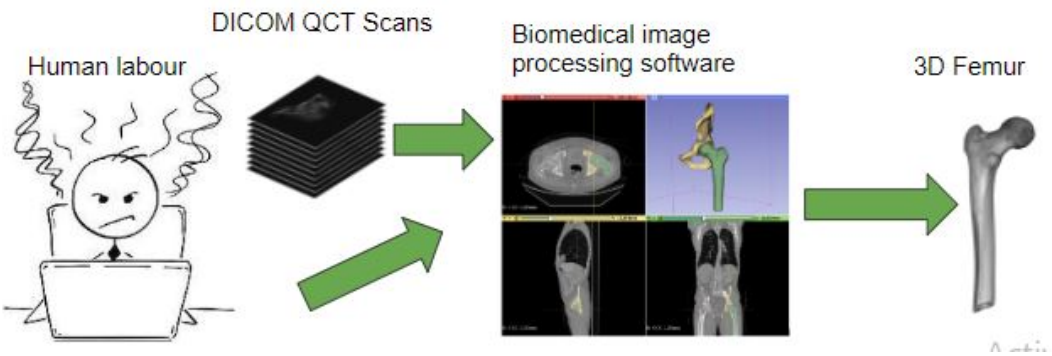


Figure 1.2: An example usage of expensive biomedical image processing software and extensive human resource involvements.

Koh et al. proposed a 3D template-based reconstruction for predicting patient-specific 3D femur geometry using X-ray and sparse CT images [5] requiring manual segmentation as well as higher reconstruction time. A model-based iterative reconstruction algorithm was proposed to work with QCT images to perform automatic segmentation of tissues to estimate material properties [30]. In addition, a gradient-based method [31] was also used for pelvic bone segmentation. However, these algorithms are very sensitive to region growing procedures. Moreover, proper thresholding is also needed to get a well segmented bone. Kalshetti et al. [32] proposed an interactive medical image segmentation framework combining mathematical morphology and GrabCut algorithm [3]. However, the performance and accuracy of this model highly depends on the distinguishing boundary of the *Region of Interest* (*RoI*) and its contrast with the background as seen in Figure 1.3.

To automate the segmentation in the 3D construction framework, an iterative algorithm but with high computational cost was used [33]. Krčah *et al.* [4] utilized statistical shape and intensity modeling that requires heavy post-processing including morphological erosion for satisfactory results. To eliminate the bottleneck, data-driven approach—Machine Learning (ML)/Deep Learning (DL) based algorithm can be incorporated in segmentation framework [34–36] to isolate proximal femur from other anatomical features.

In recent years, a few DL based frameworks have been implemented to segment proximal femur [37–39]. An automated 3D U-Net framework with a sparsely annotated training set was implemented to segment volumetric images of kidneys [40]. However, this framework is not able

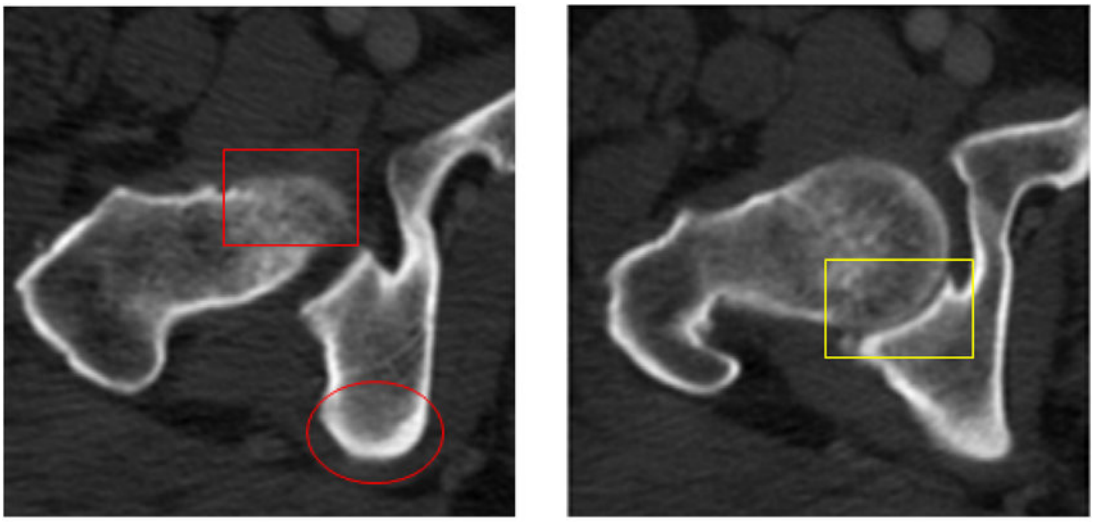


Figure 1.3: An example of low contract in femur segmentation process between the Region of Interest (RoI) and background [2].

to be generalized either for new patients or for hard tissues, especially when segmenting a bone directly from QCT images. Recently, *Convolutional Neural Network* (CNN) based 3D *U-Net* architecture has been developed for pelvic bone segmentation from Dual Energy CT (DECT) images [41]. U-Net [42], CNN based model, and data augmentation techniques [43] have shown promising performance in research for automated bone segmentation from QCT images. The performances of these frameworks are limited, because segmenting a femur surrounded by pelvic bone with similar bone density is difficult to comprehend without manual intervention. However, a performance enhancement was observed in femur segmentation with Dice Similarity Coefficient (DSC) more than 95%, utilizing Edge Detection, Deep CNN, Conditional Generative Adversarial Network (CGAN) [44–46]. Moreover, the patient-specific performance of these frameworks was not explicitly evaluated especially for a new, unseen patient outside of the study cohort.

1.3 Contribution

We have summarized the contributions of our research study as follows.

- We have proposed a semi-supervised Deep Learning based approach where we have utilized a CNN-based 3D U-Net model consisting rectangular kernel for proximal femur segmentation. This technique adds to the fast learning of the model. We name our framework *SSDL*.
- We have trained our proposed SSDL femur segmentation model as patch-based that handles the memory constraint problem.

- We have selected a semi-supervised learning model that needs less data labeling. We have only annotated geometrically complex QCT slices of the proximal femur. Our SSDL model is able to identify and isolate other less geometrically complex proximal femur bones.
- Our proposed framework segments the femur and preserves DICOM metadata from unseen patients. This method preserves spatial data to reconstruct the 3D femur from QCT slices.
- Our framework SSDL is able to reconstruct comparatively accurate proximal femur using spline polynomial interpolation. Here, the 3D reconstructed femur using SSDL, has solid femoral head along with a hollow shaft according to the femur's original geometric properties.

1.4 Organization of The Thesis

This thesis is organized as follows. Chapter 2 discusses the relevant work of femur segmentation and reconstruction. Chapter 3 describes the problem domain, data acquisition, data pre-processing and the details of segmentation. Chapter 4 provides the model training, and the steps of 3D reconstruction of the femur bone. Chapter 5 describes the details of the results and analysis. Finally, Chapter 6 provides the conclusions with the future direction of this work.

Chapter 2

Background Study

There have been some methods to design the automated proximal bone segmentation framework. However, most of the studies for bone isolation and reconstruction can be divided basically into three different groups described as follows.

- Traditional segmentation and reconstruction approaches, and
- ML/DL based segmentation and reconstruction approaches.

In conventional segmentation approaches, researchers have utilized traditional segmentation algorithms and thresholding to segment the femur [47]. Vasilache *et al.* [31] proposed a gradient-based method for pelvic bone segmentation. Their gradient based segmentation method combined *Wavelet processing*, *Laplacian filtering*, *Morphology* operations, a series of region growing techniques to create an automated segmentation system. However, these algorithms are very sensitive to region growing procedures and an optimum threshold value is required to get a well segmented bone. Stawiaski *et al.* [3] proposed a graph cut based algorithm where they used the *Watershed* method [48] along with other processing including Wavelet processing, Laplacian filtering, etc. for liver segmentation from CT images. Their model used a region adjacency graph of a watershed transform for segmentation where the process of their algorithm is shown in Figure 2.1.

Krčah *et al.* [4] proposed a bone segmentation method without requiring any prior shape template. They proposed a graph-cut method (shown in Figure 2.2) that produced a binary volume where all bone voxels were labeled. However, due to narrow inter-bone spacing, this procedure did not completely segment the femur, leading to leakage into adjacent bones such as the pelvis, tibia, and patella. Their utilized statistical shape and intensity modeling required heavy post-processing, including morphological erosion, for satisfactory results. Experimental results indicated that adjacent bones were typically connected to the femur by only a few voxels in such cases. To address this voxel connection issue, a post-processing step was employed. In this step, individual

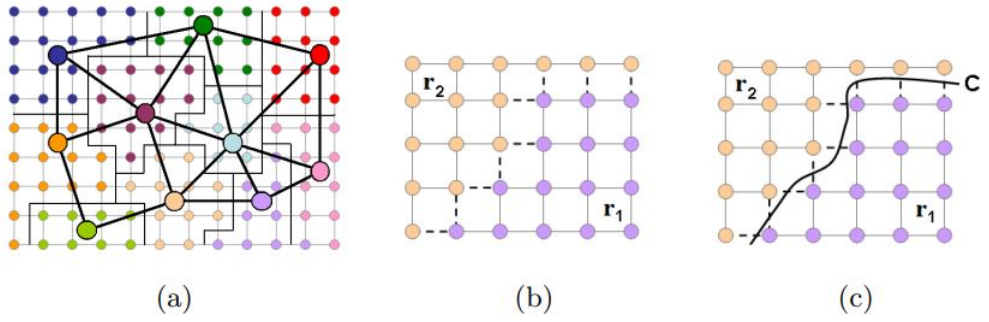


Figure 2.1: A region adjacency graph cut algorithm for liver segmentation from CT images [3]. (a) A region adjacency graph, (b) The set of nodes of the pixel graph considered to compute boundary properties between two regions, with a V4 adjacency system, (c) A curve crossing the edges of the boundary between two regions r1 and r2.

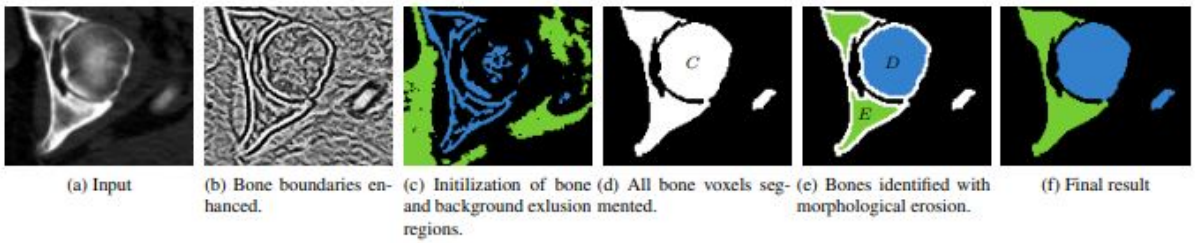


Figure 2.2: The proposed method by Krčah *et al.* [4] illustrated on an axial slice of the acetabulam-femoral joint of the input 3D CT image.

bones were identified, and the femur was determined as the largest connected component in the final volume.

Koh *et al.* [5] proposed a 3D template-based reconstruction for predicting patient-specific 3D femur geometry using X-ray and sparse CT images requiring manual segmentation as well as higher reconstruction time. The researchers combined *shape priors*, *statistical shape modeling*, and *image registration techniques* for 3D reconstruction. They leveraged the information from both X-ray and sparse CT images for femur with higher precision. This in turn, reduced the need for extensive CT scanning but increased the demand for higher reconstruction time.

Other authors started using concurrent segmentation processes that mainly involved statistical and semi-automated techniques such as atlas-based segmentation [6, 7, 27]. The authors created an anatomical atlas which is a reference image that provides the pre-defined representation of the femur shape and appearance. Registration involves aligning the atlas to the patient's image, differences in size, position, and orientation. Then, they utilized these atlas to segment the femur bone in medical images. The segmentation algorithm transfers the pre-defined femur contours from the atlas to the patient's segmented femur image by outlining the boundaries. However, for this semi-automatic method to segment the femur bone successfully, the researchers need to utilize a statistical representation of the femur derived from a huge number of training data with various shapes, appearances, orientations, etc.

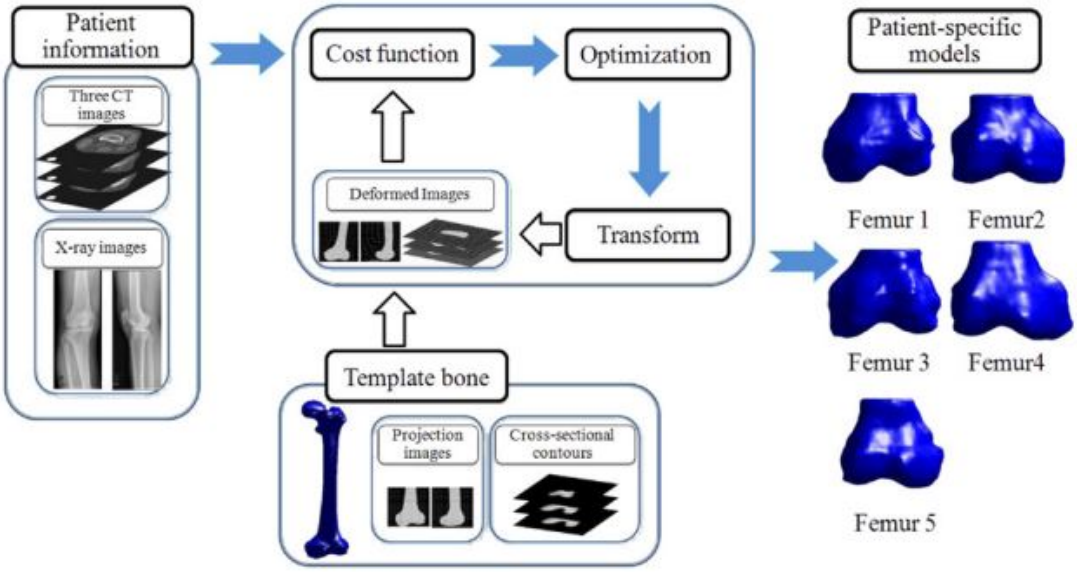


Figure 2.3: B-spline free-form deformation process with two X-ray images and three CT images proposed by Koh *et al.* [5].

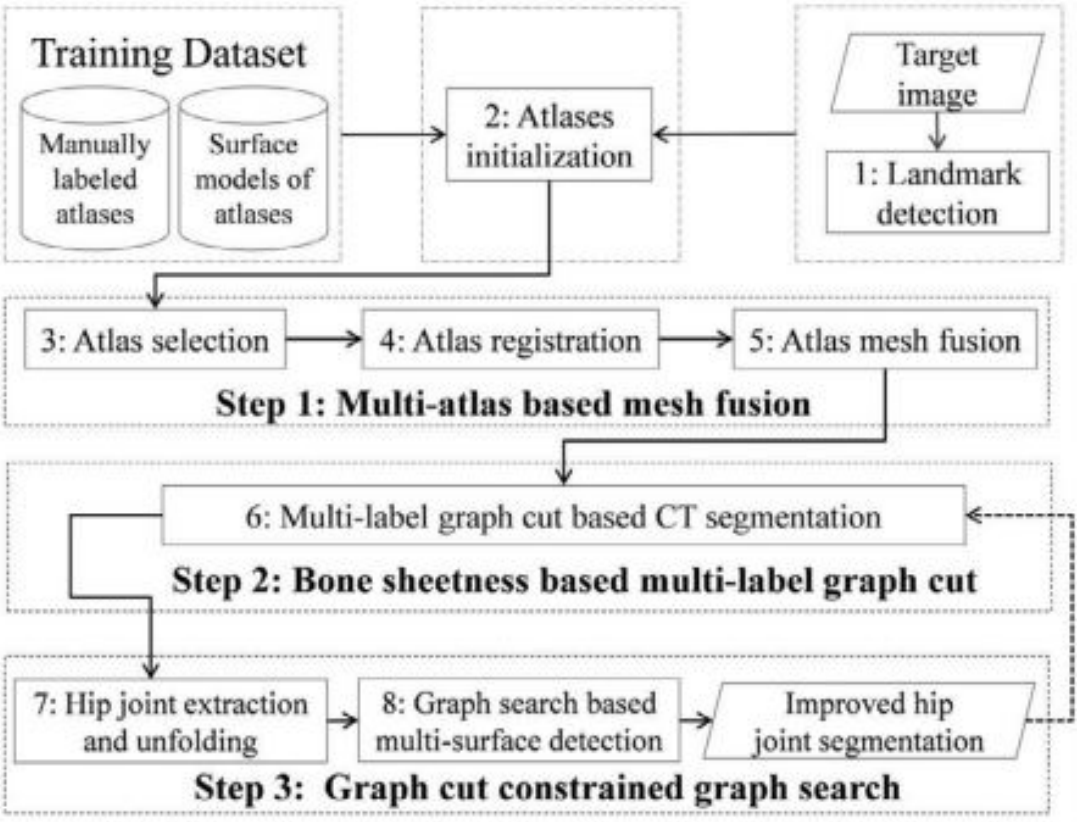


Figure 2.4: Graph cut constrained graph search and atlas-based femur segmentation method proposed by Chu *et al.* [6].

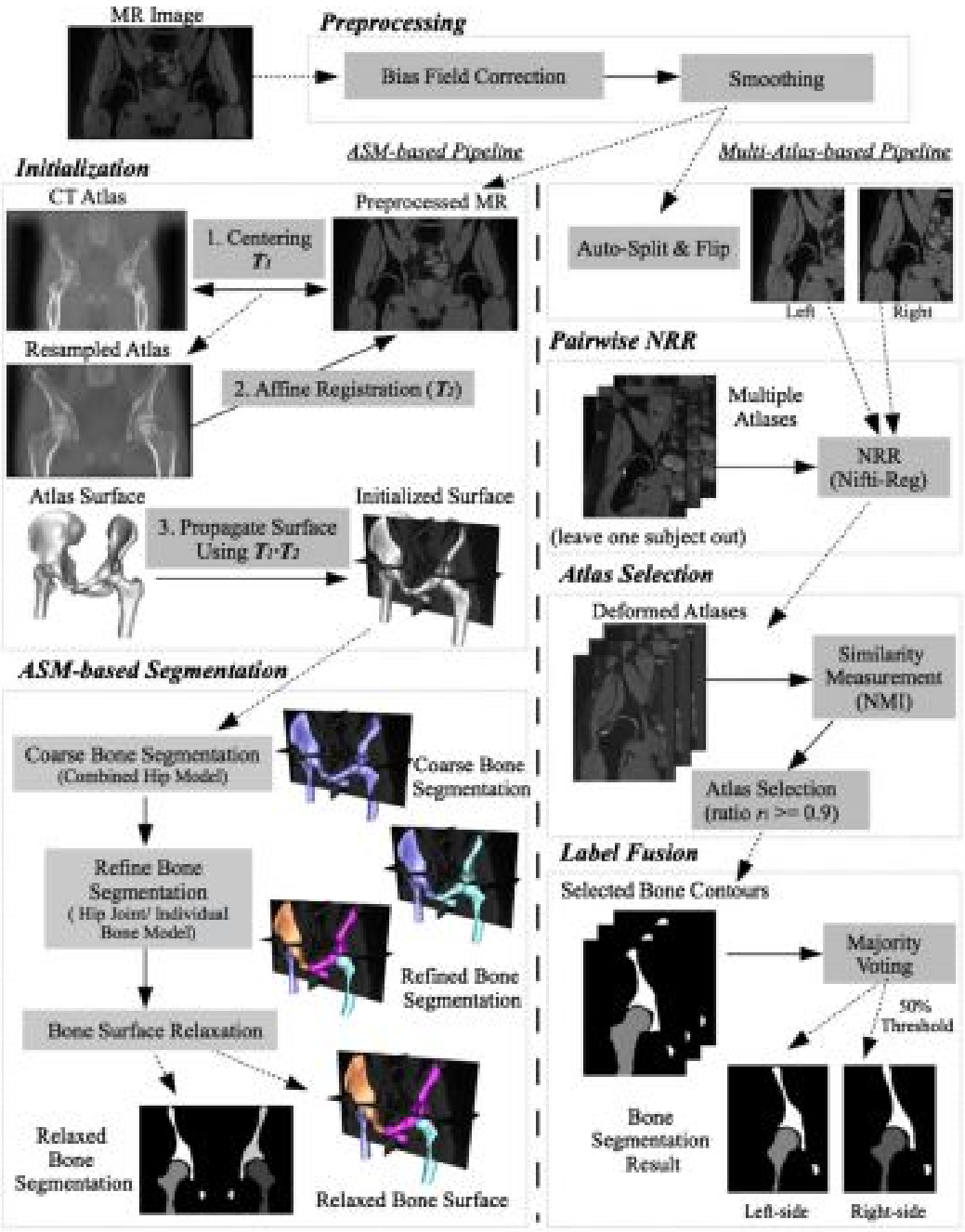


Figure 2.5: Atlas-based femur segmentation method proposed by Xia *et al.* [7].

Along with atlas based segmentation, graph-cut approaches [8, 28] and statistical shape models [28] started to become popular as traditional segmentation methods. In graph-cut approach, Xia *et al.* [8] arc-weighted graph searching techniques for automatic femur segmentation. They extracted the femur and acetabulum regions using thresholding and morphological operations. They created nodes from the extracted voxels and the connections between the voxels were named edges. Then, a graph structure is constructed by these nodes and edges based on these extracted regions. Their arc-weighted graph approach considered both intensity and spatial information. Their algorithm started from seed points within the femur region, and iteratively expands the search based on the weighted graph. The searching process incorporates constraints based on intensity and spatial relationships to segment the femur. However, this process is very computationally inefficient and based on the seed point, the femur segmentation becomes more difficult.

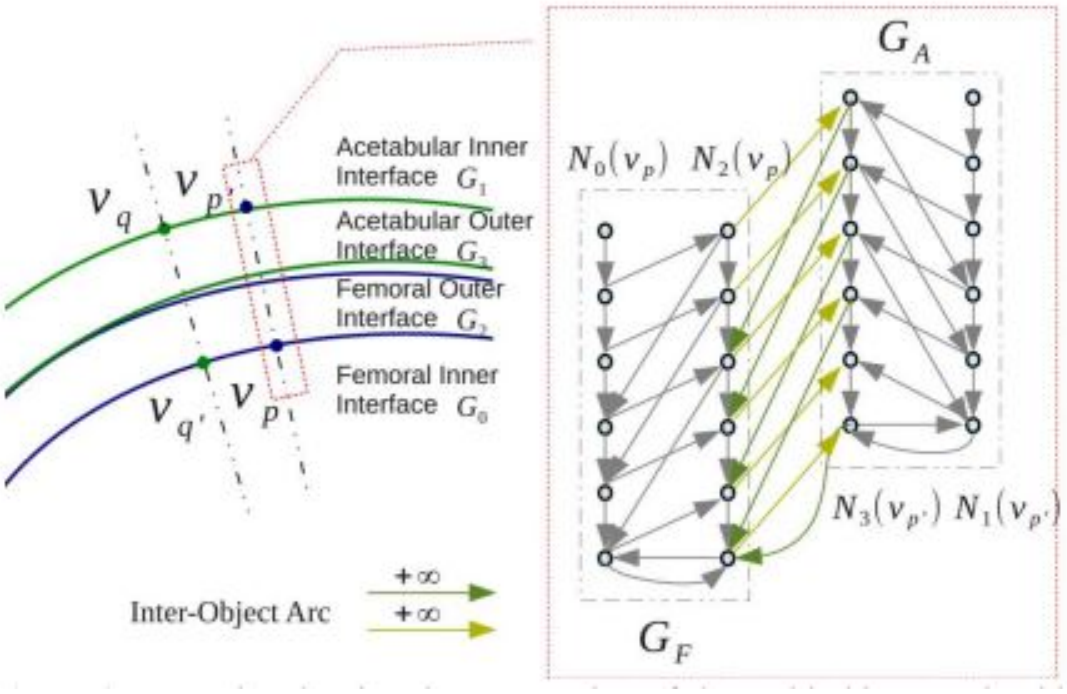


Figure 2.6: Hip cartilage segmentation from 3D MR images using arc-weighted graph searching by Xia *et al.* [8].

Later, researchers like Chandra *et al.* [28] used both graph cut algorithm and statistical shape model for femur segmentation. They kept the graph cut approach intact and added a statistical shape variation from manually segmented femur bone. This method improved the accuracy of femur segmentation with an active localized shape model. However, the previous problems of heavy computation, a huge training set of the manually segmented femur, etc., were still not solved from this approach.

Some researchers then studied similar traditional algorithms such as primitive shape recognition methods [26], active model [9, 29], model-based iterative reconstruction [30, 32] which usually

required time-consuming user monitoring and significant manual intervention. Moreover, the performance and accuracy of these models highly depended on the distinguishing boundary of the region of interest and its contrast with the background.

Malusek *et al.* [33] developed a novel iterative reconstruction algorithm named *DIRA*. The algorithm utilized model-based iterative reconstruction, automatic segmentation, and multi-material tissue decomposition. Although they showed a fast convergence within five iterations while considering noise levels, the algorithm was computationally complex for using material doublets and triplets. Therefore, for their automated femur segmentation in the 3D construction framework, the iterative algorithm had a high computational cost.

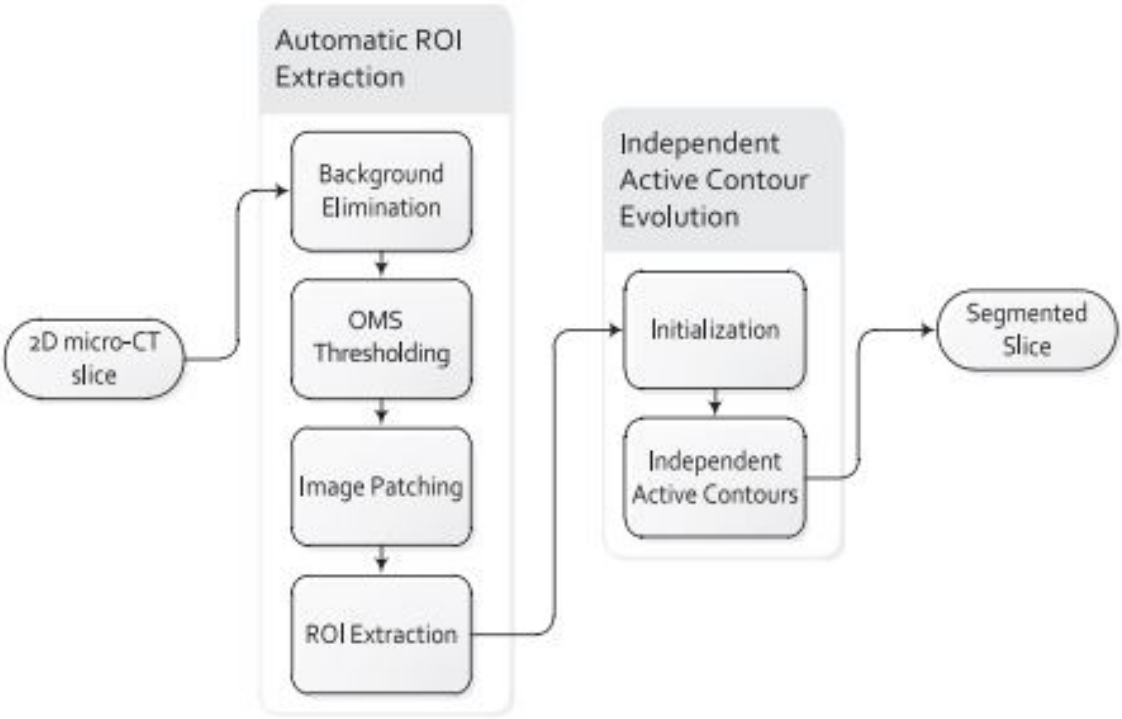


Figure 2.7: An independent active contours segmentation framework for bone micro-CT images [9].

Several similar studies [9, 29] have been conducted using traditional segmentation algorithms. Within this research group, the majority of studies focus on rigid algorithms, where specific algorithms are applied to specific regions of the body. Femur segmentation were primarily achieved through histogram matching and thresholding segmentation techniques. Some studies have taken an iterative approach to improve their algorithms, resulting in time-consuming processes. Furthermore, certain studies [33] have employed region growing or hole filling algorithms to extract compact bone, but they faced challenges when dealing with cancellous bones. Algorithms that lack prior information from human data have minimal chances of success. Most of these approaches have been tailored to individual patients, requiring ad-hoc corrections by experts to obtain better results.

In recent years, most of the femur segmentation methods have been developed using Machine Learning (ML) and Deep Learning (DL) based segmentation techniques [37–39]. These ML/DL based approaches need an abundance of annotated ground truth data for training. Thus, data pre-processing steps play a crucial role. Data augmentation techniques, such as applying artificial deformations, also needed to be employed to increase the training data for improved model performance. Hence, Çiçek *et al.* [40] proposed a semi-supervised U-Net segmentation model for soft tissue (such as, liver, kidney, brain, etc.) using sparse volumetric annotation. The authors [40] proposed a data augmentation strategy to generate additional training data by applying random transformations to the available annotations. This augmentation technique helped the data scarcity problem.

However, in the bio-mechanic field, data augmentation is frowned upon due to the uncertainty and difference between the augmented data and real data [13, 41, 43]. Moreover, this framework was not generalized for new patients and could not segment hard tissues such as bones. In segmentation using DL techniques, the U-Net model is widely popular approach [37, 42, 43]. In recent years, a few DL based frameworks have been implemented to segment proximal femur [37, 39]. Bjornsson *et al.* [37] utilized *Deep Convolutional Neural Network* (DCNN) to segment proximal femur. To train the model, a pre-processing step was employed to normalize and standardize the CT images. Data augmentation techniques, such as random rotations and scaling, were applied to augment the training dataset and improved the model's performance. A similar approach can be seen by other researchers like Sanchez *et al.* [41] where they used a CNN based 3D U-Net architecture for pelvic bone segmentation from Dual Energy CT (DECT) images.

Likewise, Deng *et al.* [45] utilized a CNN method for femur segmentation where the network was trained on a large dataset of CT images with manually labeled proximal femur regions. A pre-processing step consistent with enhancing the image quality of CT images and normalizing intensity was utilized for training the model. Data augmentation techniques, with random rotations and scaling, were applied to improve the model's performance. U-Net [42], CNN based model [37, 41], and data augmentation techniques [43] have shown promising performance in automated bone segmentation from QCT images. All of these frameworks follow similar application techniques to segment bone from medical images. They utilized a huge amount of manually segmented and annotated bones for training their model. Along with large training data, an extensive data augmentation process is also utilized for creating diverse training data of different shapes, orientations, appearances, etc. However, allmost all of the frameworks with these processes are based on supervised learning. While, a semi-supervised DL approach could use less annotated data for training, researchers utilized ad-hoc correction for better performance [40].

Performance enhancement was observed in femur segmentation with *Dice Similarity Coefficient* (DSC) more than 95%, utilizing edge detection, *Deep CNN*, *Conditional Generative Adversarial*

Network (CGAN), *Convolutional Auto Encoder*, and *K*-means clustering [44, 46]. In [46], the researchers classified all femoral CT slices based on their spatial locations using a CNN. The slices were then divided into upper, middle, and lower segments of the femur head. Then, CGAN was employed to segment the femur head areas for each segment. In addition, a Convolutional Auto Encoder was used to extract features and reduce the dimensionality of the femur head. Finally, K-means clustering was applied for the unsupervised classification of early osteonecrosis of the femoral head. In [44], Chen *et al.* utilized edge detection tasks and multi-scale features fusion to enhance the 3D feature of femur segmentation. The edge detection task addressed the issues of narrow joint space and weak femur boundary in femur segmentation. The multi-scale features fusion module improved the segmentation by providing both local and global contexts, effectively dealing with the challenges posed by variations in leg postures, femur shape, and density. Despite the progress made, it is important to note that the patient-specific performance of these frameworks has not been explicitly evaluated, especially for new patients who were not part of the original study cohort. Further research and validation are necessary to assess the generalizability and robustness of these segmentation frameworks in real-world scenarios with different patient populations. The existing frameworks for femur segmentation encounter limitations, particularly when dealing with the challenge of segmenting a femur that is surrounded by pelvic bone, which has similar bone density. These frameworks often require manual intervention to achieve satisfactory results.

Recent advancements have shown promising performance enhancements in femur segmentation. However, there are many drawbacks associated with these approaches. Firstly, a substantial number of CT scans of patients are needed as training data for the supervised models. Secondly, artificial augmentation techniques may render the model unreliable in real-life scenarios. Additionally, a significant amount of manual labor is required for expert annotations. However, it is also challenging to apply unsupervised models, such as clustering, in this context [47]. CT data loses information when converted into traditional image formats for machine learning purposes.

There have been several studies conducted on QCT/FEA (Finite Element Analysis) methods and their applications. These studies primarily aim to estimate the strength of the proximal femur and predict stresses and strains within the bone, considering the location and direction of impact forces. They often focus on bone geometry and bone markers. However, it is worth noting that these studies require the metadata of CT files, which is typically discarded by most segmentation frameworks. The primary applications of these studies include assessing hip fracture risk and studying conditions such as osteoporosis.

In this field, research was conducted by Sheehan et al. [13]. In that research, they focused on the reconstruction of 3D fracture fragments using hip CT scans. The objective of their study was to advance the evaluation of injury details and improve preoperative planning for intertrochanteric femoral fractures (IFF). By utilizing QCT and FEA techniques, the researchers aimed to enhance the understanding of IFF and develop more effective treatment strategies.

Chapter 3

Problem Domain

In this chapter, we provide the problem domain, preliminaries about data set, annotation process, and the segmentation.

3.1 Data Acquisition

The QCT images (Figure 3.1) used in this study have been obtained using a SIEMENS S5VB40B CT scanner (Siemens Medical Solution, Malvern, USA) with acquisition and reconstruction parameters of 120 kVp and 244 mAs, respectively [23, 49]. To ensure an accurate estimation of BMD and correct scanner drift, a calcium hydroxyapatite calibration phantom (Mindways Inc., Austin, TX, USA) was mounted during imaging. The clinical dataset was acquired with two types of thickness series, fine and coarse with slice thicknesses of 1 mm and 1.5 mm, respectively, in accordance with the center's imaging protocol. The QCT image dataset of the patients in DICOM format was previously obtained removing all personal information from the Great-West Life PET/CT center in Winnipeg, Canada following the institutional guidelines. The DICOM format combines image data with metadata that typically describes a patient, imaging procedure, and spatial referencing information, which are essential to reconstruct as well as to conduct clinical and morphological studies of the femur [50].

In this study, a group of 18 anonymous adults, consisting of 10 male and 8 female were selected. The dataset comprised a total of 6,789 high-resolution QCT slices in DICOM format. The average age of male and female patients was 66.0 ± 8.012 and 67.85 ± 5.265 years, respectively. The average weight of the male and female patients was 82.187 ± 7.829 and 69.7 ± 7.518 kg, respectively. Each QCT image contains an in-plane resolution of 512×512 pixel array with approximately $1 \times 1 \times 1\ mm^3$ of voxel size.

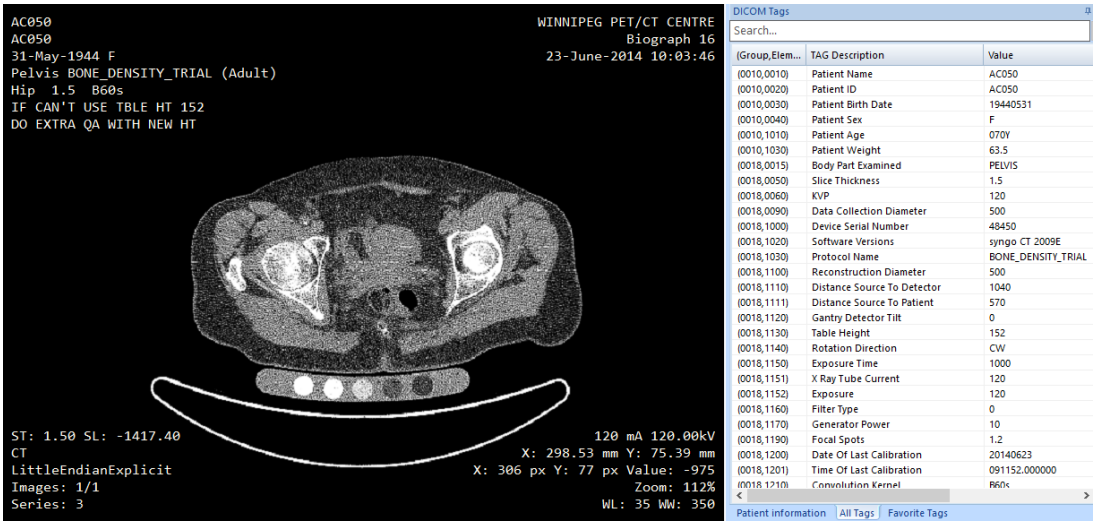


Figure 3.1: An example of a QCT image in DICOM format.

3.2 Annotating Data

In this phase, QCT slices closure to the femoral head were primarily annotated using MIALab - a free medical image analysis toolkit (Linköping University, Sweden) [38]. In hip joint near the proximal end of the femur, the femur head and acetabulum form a ball and socket joint, having nearly identical Hounsfield Units (HU) that measure bone density [51]. Consequently, differentiating the femur from the pelvic bone based on bone density alone is challenging. To address the problem, we selected a minimum number of slices in the proximity of the hip joint for labeling.

To create the training dataset for this study, between 88 and 178 QCT slices were required and selected for annotation to identify the femur profile at the proximity of femoral head and acetabulum. The variation of slice numbers was attributed to the differences in femur length primarily varying with patients' height. The annotation was performed in a semi-automated process, where a femur boundary in a slice was first segmented along the femur profile via built-in interactive region selection option. However, annotated slices were examined carefully in consultation with an experienced orthopedic surgeon for accuracy and corrected manually as necessary. This semi-automated annotation process ensured that the resulting annotated dataset was of high quality and suitable for use in training a DL based model for femur bone segmentation. In this annotation process (Figure 3.2), using the labeled data, binary masks were obtained to represent the area of the image labeled as true (i.e., contained femur bone) or false (i.e., did not contain femur bone).

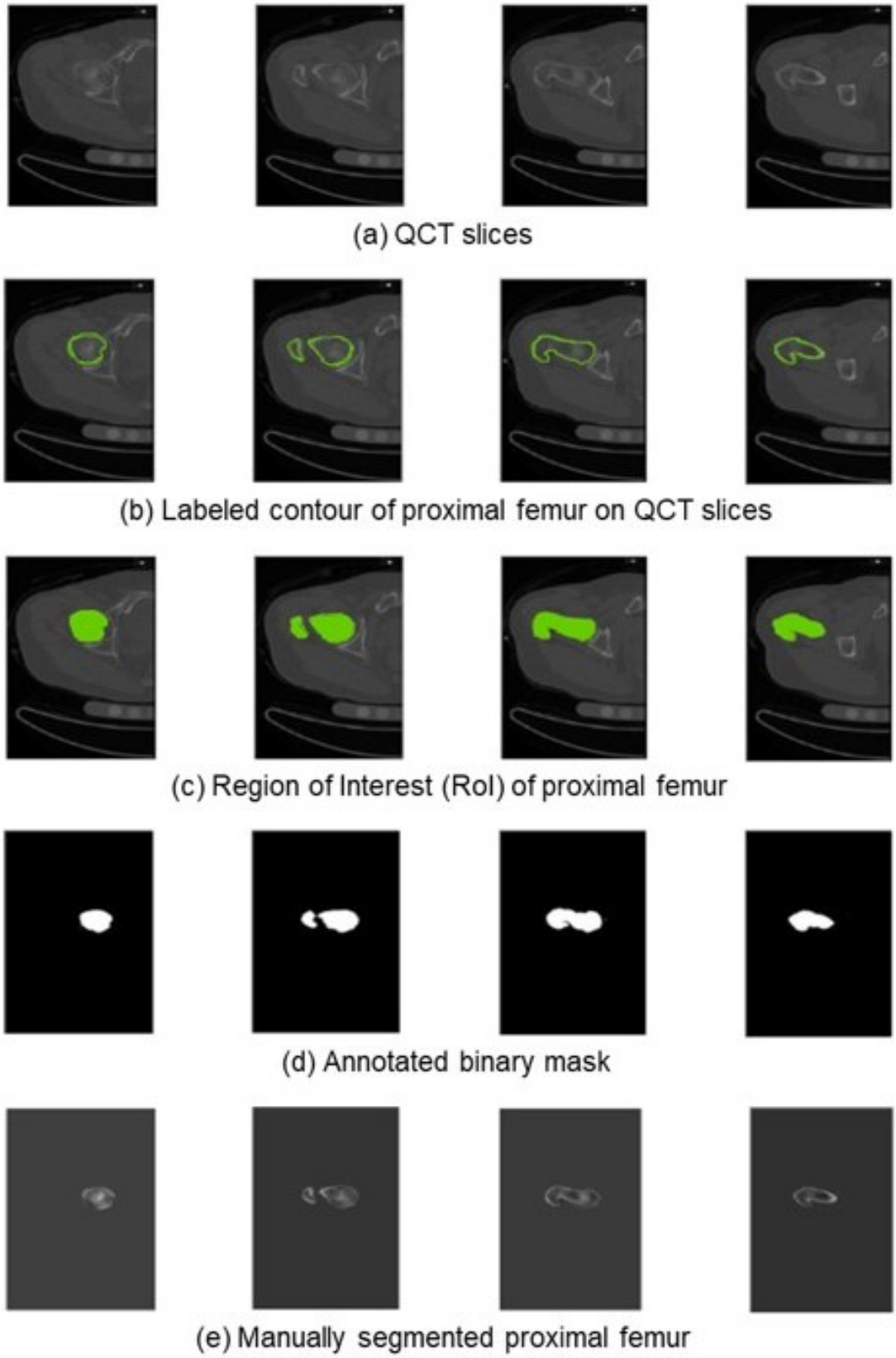


Figure 3.2: Example of annotation process for segmenting femur. (a) QCT slices belonging to different locations, (b) Labelled contour of the femur (c) Region of Interest corresponds to the contour profile, (d) Binary mask, and (e) Segmented femur.

3.3 Data Pre-Processing

In the pipeline of our femur segmentation and 3D reconstruction framework, we began by performing pre-processing on DICOM data. This involves separating the metadata and pixel values from the QCT images, which serves as the input for the SSDL model, as depicted in Figure 3.3. Within this pipeline, the framework takes the QCT images and passes them to Algorithm 1. This algorithm is responsible for obtaining a suitable input array for the *Deep Learning* (*DL*) training process.

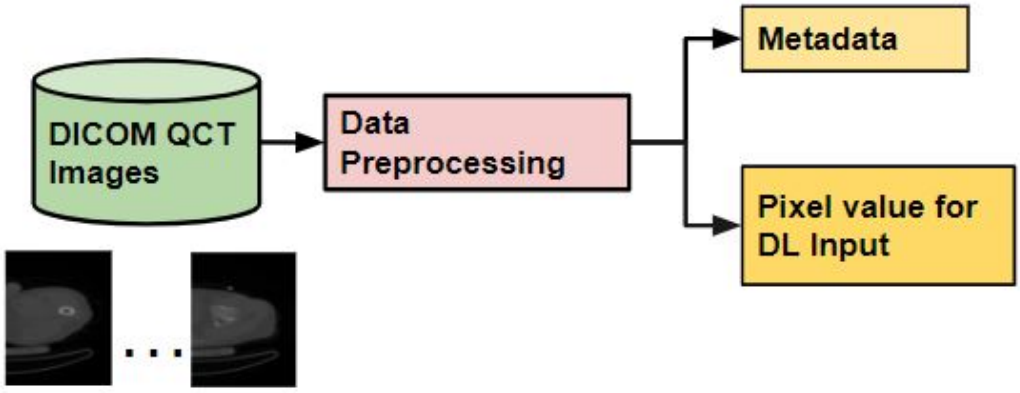


Figure 3.3: Pipeline for DICOM based QCT image processing.

The DICOM dataset was pre-processed in several steps (Figure 3.4) to facilitate data annotation and for developing training dataset. However, in this work, we have kept our pre-processing as minimal as possible, while working with metadata. Here, metadata describes the patient, the imaging procedure, and the spatial referencing information as mentioned earlier. In the pre-processing step, at first, the DICOM images have been split into half to separate left and right femurs. After splitting, each QCT image contains an image matrix of 512×256 pixels with approximately $1 \times 1 \times 1\ mm^3$ of voxel size. The number of images has been doubled after the splitting. A total number of 13,578 QCT images have been considered for the proposed framework. However, the unwanted parts of the images, such as CT bed, CT phantom, muscles, and fat, etc. have not been needed to be removed manually in this pre-processing as our proposed DL based framework can discard those unwanted parts from the DICOM images. Secondly, the training data has been batched according to their instance numbers in the QCT slices so that the correct DICOM images can be trained against their annotated labels. After that, we separate the image and text metadata since the adopted U-Net based model that would be trained only on image data. We call our framework *Semi-Supervised Deep Learning* (*SSDL*). It should be noted that, the metadata is required for the 3D reconstruction of the proximal femur. While separating the image and textual metadata, we have correctly embedded the instance number of the QCT slices from the metadata to each DICOM image. It is to be mentioned that, extreme care has been taken to avoid any mismatch of metadata and image during re-encoding the metadata with

segmented images using the instance number. Thirdly, the pixel value of DICOM images have been converted into integer16 based pixel array. After that, the pixel values have been normalized for training the SSDL model [52]. Finally, DICOM pixel array has been stored and converted into DL readable image channels of $512 \times 256 \times 1$.

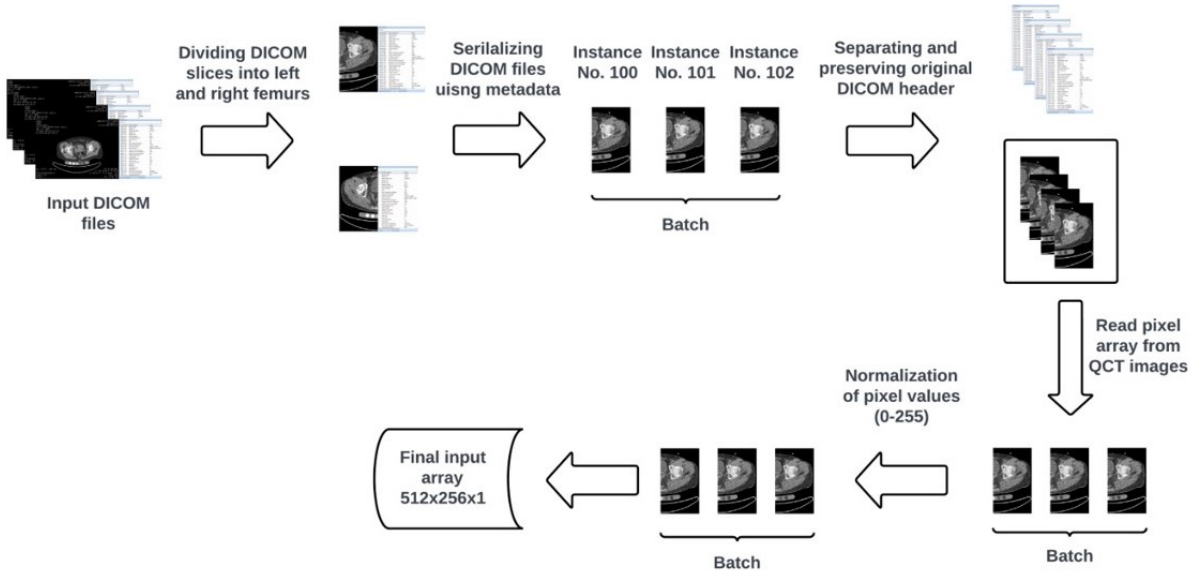


Figure 3.4: Workflow of data pre-processing steps to separate image and text metadata from the input DICOM dataset.

We start by storing all of the QCT files. To ensure compatibility with the SSDL model's training, we set the image height and width to 512 and 256, respectively, for a single channel. Next, we create a patient-specific zero array with a size equal to the total number of QCT images. This array is then populated with the processed data from the QCT images. Subsequently, we proceed to read the metadata of the QCT files for our data processing. In the while loop, we divide the QCT files into left and right until all the files have been successfully categorized using the *instance number*. This separation is important for subsequent analysis and segmentation of the femur. We also expand the dimension of the pixel array to add the pixel array of the next QCT images by following the Instance number. Additionally, we normalize the newly generated QCT pixel array, ensuring that the values were scaled between 0 and 1. This normalization step improves the convergence time of the SSDL model's training process.

The data processing Algorithm 1 presents several steps shown in Figure 3.4. In Line 7, we start our *while* loop for total QCT scans for each patient starts and read the QCT scans in Line 8. We have used from Lines 7 to 12 to serialize and divide the QCT scans into two parts: left and right femur scanning. From Lines 13 to 18, we have separated QCT image and metadata for our research experiment. Our normalization process starts from Lines 19 to 21. Here, we normalize the QCT array by using the patient-specific maximum and minimum pixel values. Since, the difference between min and max pixel values is not equal for all patients, we have used both information. We also used this normalized pixel array as the input array for SSDL.

Algorithm 1 Processing QCT images for training SSDL model.

Require: QCT images in DICOM format
Ensure: Input array in $512 \times 256 \times 1$

$StoreFile \leftarrow FolderPath$
$Total\_QCT \leftarrow length(StoreFile)$
$img\_Height \leftarrow 512$
$img\_Width \leftarrow 256$
$img\_Channels \leftarrow 1$
$ARR\_Train[Total\_QCT] \leftarrow \underbrace{\begin{Bmatrix} 0.0 & \dots & 0.0 \\ \vdots & \ddots & \vdots \\ 0.0 & \dots & 0.0 \end{Bmatrix}_{|img\_Height| \leftarrow Int16}}_{img\_Width}$
while $QCT < Total\_QCT$ **do**
  $DS \leftarrow Read\_DICOM(QCT)$
  $InstanceNo \leftarrow DS[InstanceNo]$
  $Image \leftarrow DS[Pixel\_Arr]$
  $Left \leftarrow DS[Pixel\_Arr] \leftarrow [0, img\_Height, 0, img\_Width]$
  $Right \leftarrow DS[Pixel\_Arr] \leftarrow [0, img\_Height, img\_Width, img\_Height]$
  for all $Left\&Right$ **do**
    $Left \leftarrow Embed(InstanceNo)$
    $Right \leftarrow Embed(InstanceNo)$
    $Image \leftarrow Resize(Image, (img\_Height, img\_Width), Constant)$
    $Image \leftarrow Image.Exapnd\_Dimensions[Axis = -1]$
    $ARR\_Train[QCT] \leftarrow Image$
    $min \leftarrow Min(ARR\_Train[QCT])$
    $max \leftarrow Max(ARR\_Train[QCT])$
    $ARR\_Train[QCT]_{norm} \leftarrow \frac{(ARR\_Train[QCT] - min)}{max - min}$
  end for
  $QCT \leftarrow QCT + 1$
end while

3.4 CNN-Based Framework

Once the data processing is complete, our next step in the pipeline involves modeling SSDL for femur segmentation, as illustrated in Figure 3.5. This step of the pipeline focuses on constructing the architecture and components of femur segmentation. The SSDL model is specifically designed for segmenting femur directly from the pre-processed data of QCT images.

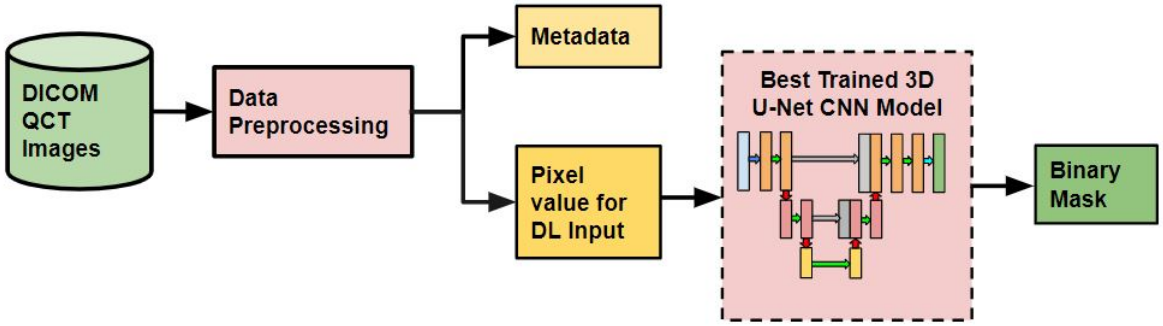


Figure 3.5: Pipeline for generating binary mask in SSDL model.

Our 3D U-Net based SSDL architecture (Figure 3.6) encodes the latent and high-level features of the input pixel array of raw QCT images and decodes the features to generate the final segmentation results in a binary pixel array of true and false. The U-Net architecture developed herein consists of six deep learning components, and two paths—contracting (*Downsampling*) and expanding (*Upsampling*) paths. The contracting path is the encoder and captures context using stacked *Convolutional* units and *Max Pooling* layers. In contrast, the expanding path is the decoder and allows for precise localization using Upsampling. Figure 3.6 shows the CNN based architecture of our SSDL model with pre-processed QCT images as the input and the generated binary mask as the output.

In Algorithm 2, we present the pseudo code for our proposed SSDL model. The model can be divided into two main parts, as previously discussed: *Downsampling* and *Upsampling*. The algorithms for these components are provided in Algorithm 3 and Algorithm 4, respectively. We prepared an *Input Layer* with dimensions matching the height and width of the QCT pixel array before the downsampling part started. We then fed the input array, with a shape of $(512 \times 256 \times 1)$, into the downsampling block utilizing convolutional layers. Following the downsampling process, we incorporate dropout to mitigate the risk of overfitting [53]. The output from this layer is subsequently fed into the Upsampling block. The Upsampling block processed the input and generated a binary mask as the output, with a shape of $(512 \times 256 \times 1)$. This mask represents the segmentation results obtained from the SSDL model.

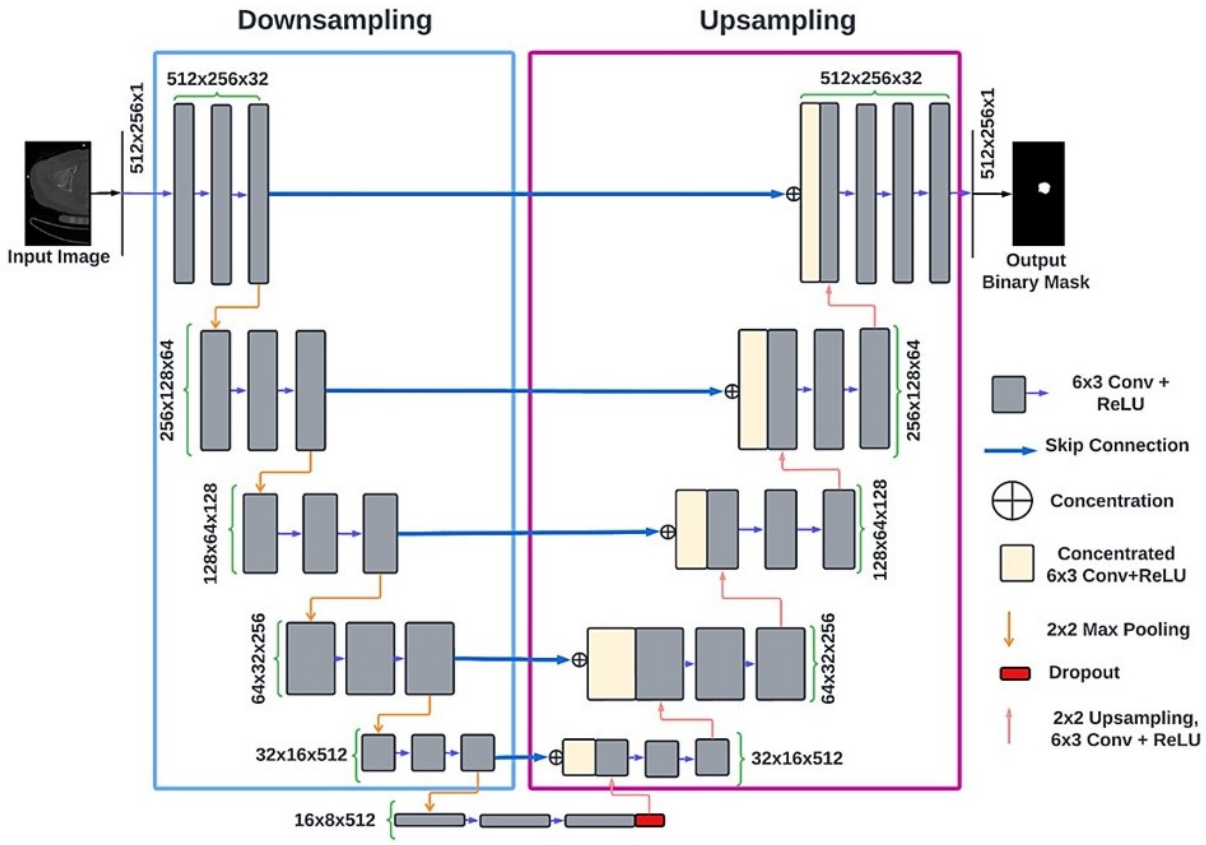


Figure 3.6: CNN architecture of end-to-end 3D U-Net based SSDL model.

Algorithm 2 Algorithm for SSDL femur segmentation model.

Require: $Input\_array[512 \times 256 \times 1] \leftarrow QCT\_images$
Ensure: $BinaryMask\_array[512 \times 256 \times 1]$
1: $img\_Height \leftarrow 512$
2: $img\_Width \leftarrow 256$
3: $img\_Channels \leftarrow 1$
4: $input\_layer \leftarrow Input((img\_Height, IMG\_Width, img\_Channels))$
5: $conv\_in \leftarrow Conv2D(ReLU, padding, kerinit))(input\_layer)$
6: $Downsampling(conv\_in)$ **(Downsampling Algorithm 3)**
7: $conv \leftarrow Conv2D\_Block$
8: $drop \leftarrow DropOut(conv)$
9: $Upsamling(drop)$ **(Upsampling Algorithm 4)**
10: $conv\_out \leftarrow Conv2D(Sigmoid, padding)(conv_{upsampling})$

3.4.1 Convolutional Layers

The SSDL model comprises two-dimensional Convolutional Layers that contain a number of filters to be applied to the feature input. It is a process where we take a small matrix of numbers as kernel, we then pass it over the image, and transform it based on the values from the filter (convolution layers).

$$(f * h)[m, n] = \sum_{j} \sum_{k} h[j, k] \cdot f(m - j, n - k) \tag{3.1}$$

Equation 3.1 is the convolution layer of $(f * h)[m, n]$ from where subsequent feature map values are calculated. In Equation 3.1, f is a 2D input pixel array and h is the kernel. The indexes of rows and columns of the result pixel array are marked with m and n, respectively. After placing the filter over the selected pixel numbers, each value of the kernel was considered and multiplied them in pairs with corresponding values from the pixel array. Finally, we sum up everything and put the result accordingly in the output feature map. The features map at location i is given in Equation 3.2 as follows,

$$F^l{}_f(i) = W^l{}_f(i)x^l(i) + b^l{}_f \tag{3.2}$$

, where $F^l{}_f(i)$ is the feature map in the ith location of the lth layer, F is the computed features map with $W^l{}_f(i)$ weight matrix of the kernel centered at the location, $x^l(i)$ is an input vector and $b^l{}_f$ is the bias. To address image shrinkage and the reduction of pixel size at the image edges during convolution, an extra border is added to the image by padding it with zeroes. This additional border around the image ensures and maintains the same size of the input and output images. In this work, we have used padding (Equation 3.3) with an odd filter dimension, and the padding width satisfies Equation 3.3,

$$p = \frac{f - 1}{2} \tag{3.3}$$

, where f is the filter dimension. In this work, convolution layers utilize the Keras kernel initializer *he normal* [54]. This initializer draws samples from a truncated normal distribution that is centered at 0. The formula for the truncated normal distribution is given by Equation 3.4, where fan_{in} represents the number of input units in the weight tensor.

$$stddev = \sqrt{\frac{2}{fan_{in}}} \tag{3.4}$$

3.4.2 Activation Function

In the neural network, an activation function plays a crucial role in handling the weighted sums between the layers in the model architecture. We used two different activation functions to observe the effects of activation layers in our SSDL model.

- ***Rectified Linear Activation Unit (ReLU):*** The activation function— ReLU activation layer works by setting all the negative numbers in the matrix to 0, keeping the non-negative numbers in the input unchanged. Since, all pixel values associated with the femur bone are greater than 0, the ReLU activation function has been found suitable for our femur segmentation process.

$$ReLU(x) = max(0, x) \tag{3.5}$$

- ***Parametric Rectified Linear Unit (PReLU):*** The activation function- PReLU [55] activation layer generalizes the traditional rectified unit with a slope for negative values. It also reduces the number of unused neurons in the network, which improves the learning speed and overall performance of the network. PReLu works as a leaky ReLU with its parametric value α (Equation 3.6). However, the difference between *Leaky ReLu* and *PReLU* is that PReLU lets the network figure out when the parameter is needed. In our case, for $x < 0$, our SSDL model's activation layer uses the parameter $\alpha = 0.02$. This way, PReLU also solves the dying ReLU problem of zero slopes.

$$PReLU(x) = \begin{cases} \alpha * x, & \text{if } x < 0. \\ x, & \text{if } x >= 0. \end{cases} \tag{3.6}$$

Both of these activation functions have been mainly used as DL methods to handle non-negative input numbers representing femur pixels and negative input numbers representing the pixels of the background.

3.4.3 Pooling Layer

One of the schemes of the pooling layer is the Max pooling layer. A Max-pooling layer operates by selecting the maximum element from the region of a feature map covered by a filter. Therefore, the output after the max-pooling layer will be a feature map containing the most prominent features of the previous feature map. Equation 3.7 shows the dimensions of output obtained after a max pooling layer where n_h, nw, nc are height, width, and number of channel in the feature map, respectively. f is the size of the filter and s is stride length.

$$MaxPool[n_h \times n_w \times n_c] = \frac{n_h - f + 1}{s} \times \frac{n_w - f + 1}{s} \times n_c \tag{3.7}$$

3.4.4 Downsampling

In this SSDL model, Max Pooling layers along with 2D convolution layers, kernel initializer "he normal" [54], Rectified Linear Activation Unit (ReLU) [52] have been used for downsampling the dimensionality of feature maps. Convolutional layers contain several filters to be applied to the feature input.

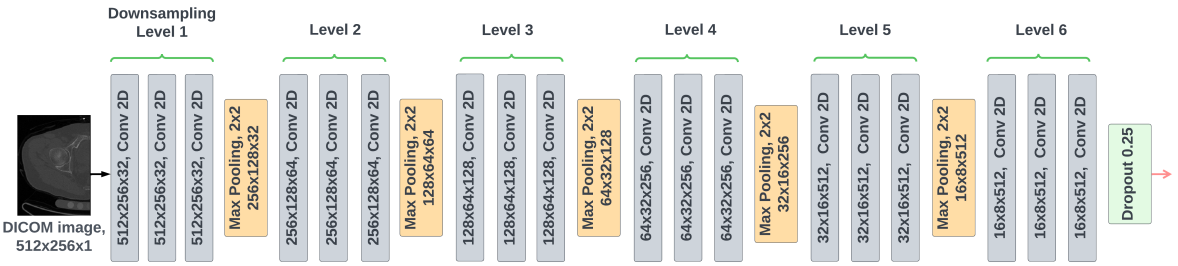


Figure 3.7: CNN architecture of downsampling of proposed SSDL model.

Increasing Speed

Since the orientation of the DICOM slices is known, and the segmentation's feature map is taller than its width, we have utilized a kernel of 6×3 matrix size in convolution layers to restrict fast learning. This rectangular kernel matrix maps each input point using 2 times more along vertical than horizontal neighbors. Given the similar bone density distribution in pelvis, our SSDL model utilizes the rectangular kernel to identify discriminative features in the acetabular regions compared to the shaft region of the femur. Specifically, the model devotes twice as much attention to the acetabular regions. Moreover, ReLU activation sets all the negative numbers in the matrix to 0, keeping the non-negative input numbers unchanged. Since all pixel values associated with the femur bone are greater than 0, the ReLU activation function is a suitable one for femur segmentation process.

Handling Memory Constraints

To address memory constraints, we have adopted a patch-based model, where each volume patch is $512 \times 256 \times 1$ voxels with an overlap of $256 \times 128 \times 64$ voxels. This patch size captures the entire femoral head, which is the most critical section of the proximal femur, and downsamples after each use of Max Pooling (here, only the integer values of the most prominent features of the previous feature map were left for patch dimensions). To address image shrinkage and the reduction of pixel size at the image edges during convolution, an extra border has been added to the image by padding it with zeroes and an odd filter dimension. This additional border around the image ensures and maintains the same size of the input and output images. During training,

volume patches have been randomly selected from the image volume and subsequently fed into the neural network with a batch size of 10. As the image patch is downsampled in the contracting path, the number of channels increases, allowing for the extraction of more complex features. In addition, we have added a dropout layer before transferring the feature maps to the upsampling path to prevent the model from overfitting [53]. Skipping connections has been also used to pass features from the contracting path to the expanding path to recover the lost spatial information during downsampling process.

By referring to Algorithm 3 and examining Figure 3.7, we can gain insight into the architecture of downsampling part of our SSDL model. This section of our segmentation model comprises six convolutional units, each accompanied by the utilization of the *MaxPooling* technique. These components construct the downsampling component of our segmentation model and extract salient features of the QCT input images.

Algorithm 3 Algorithm for *Downsampling* in SSDL model.

Require: $Input\_array[512 \times 256 \times 1] \leftarrow QCT\_images$
Ensure: $BinaryMask\_array[512 \times 256 \times 1]$
$img\_Height \leftarrow 512$
$img\_Width \leftarrow 256$
$img\_Channels \leftarrow 1$
$input\_layer = Input((img\_Height, img\_Width, img\_Channels))$
$ReLU \leftarrow max(Input, 0)$
$conv1 \leftarrow Conv2D(ReLU, padding, ker\_init))(input\_layer)$
$conv1 \leftarrow Conv2D\_Block1$
$pool1 \leftarrow MaxPooling2D(conv1)$
$conv2 \leftarrow Conv2D\_BLock2$
$pool2 \leftarrow MaxPooling2D(conv2)$
$conv3 \leftarrow Conv2D\_BLock3$
$pool3 \leftarrow MaxPooling2D(conv3)$
$conv4 \leftarrow Conv2D\_Block4$
$pool4 \leftarrow MaxPooling2D(conv4)$
$conv5 \leftarrow Conv2D\_Block5$
$pool5 \leftarrow MaxPooling2D(conv5)$
$conv6 \leftarrow Conv2D\_Block6$
$drop \leftarrow DropOut(conv6)$

3.4.5 Upsampling

Upsampling path in our 3D U-Net based SSDL architecture decodes downsampled pixel array. We have utilized the 2D upsampling layers along with the convolutional layers to create the decoding path of our end-to-end U-Net model. At each step in the decoder, a skip connection allows for the concatenation of the output of the transposed convolutional layer with the corresponding feature map from the encoder. Skipping connections tackle *vanishing gradient*

problems by using uninterrupted gradient flow from the first convolutional layer to the last convolutional layer [56]. Features have been passed from the encoder path to the decoder path. Hence, skipping connections can recover any lost spatial information during downsampling, and stabilize gradient updates in deep architectures. Concatenative skipping connections ensures feature risibility of the same dimensionality from the earlier layers. The convolutional layer is subsequently able to produce a more precise output based on this information.

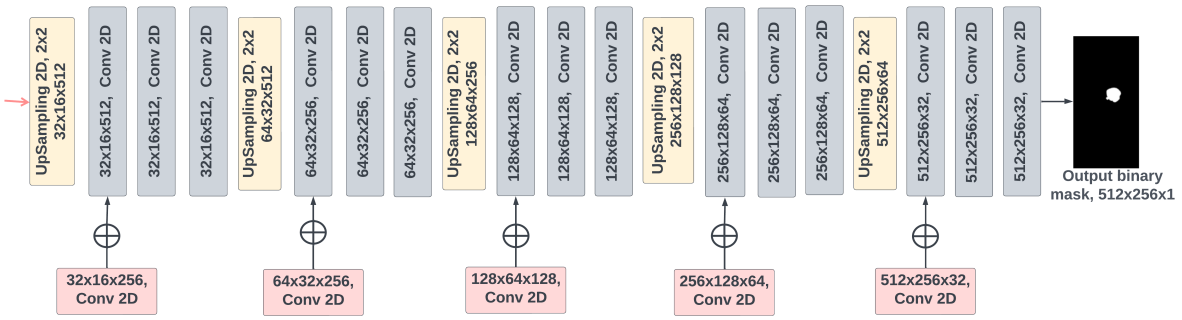


Figure 3.8: CNN architecture of upsampling of proposed SSDL model.

By referring to Algorithm 4 and examining Figure 3.8, we can gain insights into the architecture of the upsampling part of SSDL model. In this stage, we aim to decode the previously encoded salient features extracted from the QCT images. To achieve this, we employ five convolutional units along with two-dimensional upsampling of the convolutional layers. This combination of components enables us to reconstruct the feature maps, allowing for the retrieval of finer details and spatial information. After the upsampling process, the resultant neurons are fed into the "Sigmoid" layer. This layer performs a *Sigmoid* activation function that maps the output values between 0 and 1. This final step produces a binary mask with probabilistic values, representing the segmentation results of the SSDL model.

3.4.6 Generating Binary Mask

Lastly, we create a U-Net output layer made of a convolutional layer with $1 \times 1 \times 1$ kernel and *sigmoid* activation function [57]. Sigmoid function can map predicted values to probabilities between 0 and 1. This layer then transfers the feature maps into a probability image with the resolution of 512 to generate the binary output mask. The output of each voxel is the probability that the voxel in the pixel array contains the proximal femur. We then round the probability value of each voxel to the nearest integer value of 1 and 0, where 1 means true for femur presence and 0 means background, respectively. Thus, we convert the output probability map into a binary segmentation mask of true and false. Equation 3.8 explians the Sigmoid function, where e is the Euler's number.

Algorithm 4 Algorithm for *Upsampling* in SSDL model.

Require: $Input\_array[512 \times 256 \times 1] \leftarrow QCT\_images$
Ensure: $BinaryMask\_array[512 \times 256 \times 1]$
$img\_Height \leftarrow 512$
$img\_Width \leftarrow 256$
$img\_Channels \leftarrow 1$
$ReLU \leftarrow max(Input, 0)$
$up1 \leftarrow UpSampling2D(drop)$
$up1 \leftarrow Conv2D(ReLU, padding, ker\_init)(up1)$
$merge1 \leftarrow concatenate(conv6, up6)$
$conv7 \leftarrow Conv2D\_Block7$
$up2 \leftarrow UpSampling2D(conv7)$
$up2 \leftarrow Conv2D(ReLU, padding, ker\_init)(up2)$
$merge2 \leftarrow concatenate(conv7, up2)$
$conv8 \leftarrow Conv2D\_Block8$
$up3 \leftarrow UpSampling2D(conv8)$
$up3 \leftarrow Conv2D(ReLU, padding, ker\_init)(up3)$
$merge3 = concatenate(conv8, up3)$
$conv9 \leftarrow Conv2D\_Block9$
$up4 \leftarrow UpSampling2D(conv9)$
$up4 \leftarrow Conv2D(ReLU, padding, ker\_init)(up4)$
$merge4 = concatenate(conv9, up4)$
$conv10 \leftarrow Conv2D\_Block10$
$up5 \leftarrow UpSampling2D(conv10)$
$up5 \leftarrow Conv2D(ReLU, padding, ker\_init)(up5)$
$merge5 \leftarrow concatenate(conv10, up5)$
$conv11 \leftarrow Conv2D\_Block11$
$conv \leftarrow Conv2D(Sigmoid, padding)(conv11)$

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.8}$$

The *mean Intersection over Union*, *Dice Similarity Coefficient* (*DSC*), *Average Precision*, *Sensitivity*, and *Specificity*, loss functions can quantify the model performance at each iteration of each epoch in order to adjust the parameters of the gradient descent algorithm during backpropagation. The parameters compute the discrepancy between the generated binary mask and the ground truth to evaluate the current performance of the model in each step. The validation set allowed us to gain insight into the performance of the model on unseen data so that hyperparameters could be adjusted accordingly. The feature maps for all the convolutional layers in the upsampling path are shown in Figure 3.8.

Chapter 4

Reconstruction of Femur

In this chapter, we describe the details of 3D reconstruction of proximal femur.

4.1 Performance Metrics

To evaluate the accuracy and robustness of our SSDL segmentation model, we have considered several performance metrics, including *mean Intersection over Union* (*mean IoU*) [58], *Dice Similarity Coefficient* (*DSC*) [59], *Average Precision* [60], *Sensitivity* [61], and *Specificity* [61]. These metrics are also used for measuring the matching between the automated segmented femur and the manually annotated ground truth of the femur. To ensure a more rigorous evaluation, we have incorporated a cross-validation process as well.

The performance indicators have been mathematically evaluated on two classes (femur and background) based upon *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), and *False Negatives* (FN). In this problem domain, *TP* measures the proposed model's ability to detect the pixel positions on the femur foreground, *TN* denotes our model's ability to detect the pixel positions on background, *FP* is the model's ability to identify pixels on femur where the background is situated, and *FN* represents our model's ability of detecting background where the femur pixels are situated. The performance indicators are described as follows.

The *mean IoU* (Equation 4.1) evaluates the accuracy of the predicted image mask by calculating the intersection between the predicted and actual images [58]. This metric assesses the effectiveness of segmentation or accuracy of detecting objects in images in segmentation models.

$$mean\ IoU = \sum_{c \in classes} \frac{TP}{TP + FP + FN} \tag{4.1}$$

DSC has been widely used in various bone segmentation applications [34,36,37]. *DSC* (Equation 4.2) measures the similarity between the predicted segmentation mask and ground truth mask by considering both TP and TN rates [59]. The numerator in Equation 4.2 represents the number

of pixels that are correctly labeled as the object of interest in both the predicted and ground truth masks, while the denominator represents the total number of pixels labeled as the object of interest in either of the masks. DSC varies between 0 and 1, with a higher value indicating a better segmentation performance.

$$DSC = \frac{2 \times TP}{(TP + FP)(TP + FN)} \tag{4.2}$$

Average Precision (Equation 4.3) is used to evaluate the efficiency of ML models in identifying positive samples [60]. Here, TP signifies True Positive and FP is noted for False Positive. The summation of TP and FP is the total number of identified positive samples.

$$Average\ Precision = \sum_{c \in classes} \frac{TP}{TP + FP} \tag{4.3}$$

Sensitivity [61] (Equation 4.4) evaluates the model's ability to correctly identify positive instances of the pixel and position of the femur in the QCT image. A higher sensitivity value demonstrates an increased segmentation ability of our adopted framework. *Specificity* [61] (Equation 4.5) metric measures the model's ability to correctly identify negative instances. A high specificity score indicates the model's ability to detect negative instances more accurately, making it an essential metric in evaluating the overall performance of ML models. Loss functions quantifies the model's performance at each iteration of each epoch to adjust the parameters of the gradient descent algorithm during backpropagation.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4.5}$$

All these parameters compute the discrepancy between the generated binary mask and the ground truth to evaluate the current performance of the model in each step.

The validation set has allowed us to gain insight into the performance of the model on unseen data so that hyperparameters can be adjusted accordingly. For 18 patients in our dataset, we have used a *Leave One (Patient) Out Cross Validation* (*LOOCV*) [62] approach, with 8 patients for training and the remaining 10 patients for testing. Among the testing patients, four patients have been known to the SSDL segmentation model training through validation whereas the other six patients have been unseen to the model's training. For model compilation, *Adam* optimizer [63] and *binary cross-entropy*, a loss function, have been used. To ensure optimal performance, with the patience of 10 iterations and an improvement threshold of 0.0001, early stopping criteria has been employed based on the maximum validation of DSC. Additionally, a model checkpoint was set based on the maximum validation of DSC. To further enhance the learning process, the learning rate has been reduced for the model based on validation loss, with a lower bound set at

0.00001, and the learning rate has been decreased by a factor of 0.2 as shown in Equation 4.6.

$$lr\_new = lr \times 0.2 \tag{4.6}$$

The *binary cross-entropy* loss function (Equation 4.7) has been used as the loss function to optimize the neural network model by calculating the difference between the predicted and true probability distributions.

$$H_p(q) = -\frac{1}{N}\sum_{N}(y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))) \tag{4.7}$$

where y and $\log(p(y))$ represent the true and predicted labels, respectively.

4.2 Testbed Description

The testing environment for our experiments involved a personal laptop computer with an Intel Core i7-7500U CPU running at 2.90 GHz, 4 MB cache, and 8GB of RAM. In addition, we utilized an NVIDIA GeForce GTX 940 MX GPU with 8094 MB of memory and a refresh rate of 60 Hz. We have chose to use the Python3 programming language [64] due to its ease of use and availability of relevant libraries. For the implementation of our model, several Python libraries have been incorporated in this work. We have used TensorFlow [65] — an open-source software library for accessible and programmable DL functions for segmentation, dataflow, and differentiable tasks, PyDicom [66], which is a pure Python package for working with DICOM files, SciPy [67] library for scientific and technical computing. We have also used *Skimage*, which is a collection of algorithms for image processing and computer vision as well as the libraries associated with *Google Colab*, the cloud-based platform for data analysis and machine learning, and *Visual Studio* Code—widely used for Python development.

4.3 Segmentation and Metadata Inclusion

After obtaining the DICOM files of the segmented proximal femur, we isolate the femur by performing a pixel-wise multiplication (Equation 4.8) using binary segmentation masks generated from our proposed SSDL model. The pipeline for the femur segmentation and metadata inclusion after generation binary mask is shown in Figure 4.1.

At first, we match the extracted metadata of DICOM files with the generated binary mask arrays based on their embedded instance numbers. To segment the femur in DICOM images, we retain the pixel value in the DICOM image if the corresponding value in the binary mask array is 1. Conversely, if the value in the binary mask array is 0, we discard the corresponding area of the

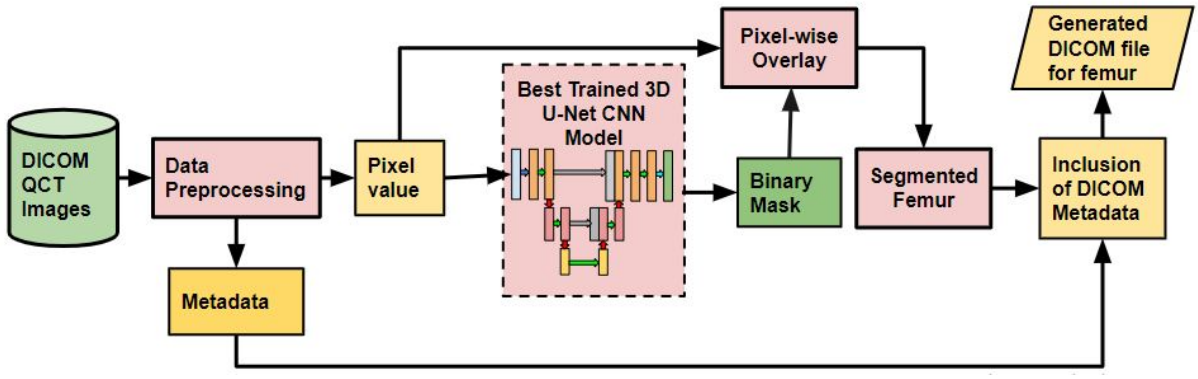


Figure 4.1: Pipeline for femur segmentation by using binary masks and metadata inclusion.

DICOM image. Equation 4.8 represents our pixel-wise multiplication methodology.

$$f(x) = \begin{cases} D(x), & \text{if } D(x) * M(True). \\ 0, & D(x) * M(False). \end{cases} \tag{4.8}$$

, where $D(x)$ represents the pixels of DICOM image array, $M(True)$ and $M(False)$ represent the values 1 and 0 in the generated binary mask array, respectively.

After performing pixel-wise multiplication, the resulting array turns into 16-bit integer format. Since our DICOM images are stored in a 16-bit format, we convert the segmented femur array to a 16-bit array and update the pixel array in the metadata to reflect the newly segmented femur bone. The metadata for DICOM rows and columns are also updated to reflect the new shape of the generated femur. Additionally, the largest image pixel and smallest image pixel values are also updated for the newly generated DICOM images. Finally, we add preserved metadata from the original DICOM files to the newly generated femur bone.

To prepare for the 3D reconstruction of the femur, we also perform some post-processing cleanup on the DICOM slices. At first, we extract the new image positions and slice locations by loading the newly generated DICOM files of the segmented proximal femur. The slice thickness can be obtained from the DIOCM metadata. Next, we remove any undesired pixels (noises) outside the range of cancellous and trabecular bones by replacing the out of bound pixel values with the HU value of air. In this denoising step, morphology masking based on HU variation is used to discard the outer variances from the slices. This eventually ensures the 3D reconstruction of the femur bone only.

Algorithm 5 outlines the pseudo code for our metadata inclusion and femur segmentation using binary masks. The algorithm begins by storing all patient-specific QCT images. It then enters a while loop that continues until all the QCT images have been processed. Within the loop, we read the original DICOM files and extract their metadata. The algorithm proceeds to compare the instance number of the original QCT image with the generated binary mask. If there is a match,

we perform a pixel-wise multiplication between the QCT image and the binary mask, effectively segmenting the femur region of interest. Additionally, we update the metadata values for the newly generated segmented femur QCT files, ensuring that the metadata accurately reflects the segmentation process. This step helps maintain the integrity of the metadata associated with the segmented femur images.

Algorithm 5 Femur segmentation by binary masking and metadata inclusion

Require: $Input\_array[512 \times 256 \times 1] \leftarrow QCT\_images, BinaryMask\_array[512 \times 256 \times 1]$
Ensure: $SegmentedFemur\_array[512 \times 256 \times 1]$
$StoreFile \leftarrow FolderPath$
$Total\_QCT \leftarrow length(StoreFile)$
while $QCT < Total\_QCT$ **do**
  $DS \leftarrow Read\_DICOM(QCT)$
  $InstanceNo \leftarrow DS[InstanseNo]$
  if $DS[InstanceNo] == BinaryMask\_array[InstanceNo]$ **then**
    $Femur \leftarrow DS[QCT] * BinaryMask\_array[QCT]$
    $FemurImg \leftarrow Int16(Femur)$
    $DS[PixelData] \leftarrow ToByte(FemurImg)$
    $DS[Rows], DS[Columns] \leftarrow FemurImg[Shape]$
    $DS[LargestImagePixelValue] \leftarrow max(FemurImg)$
    $DS[SmallestImagePixelValue] \leftarrow min(FemurImg)$
    $SegmentedFemur\_array \leftarrow DS$
  end if
end while

4.4 Reconstruction

In the final step of this workflow, 3D femur is reconstructed from the segmented CT slices preserved in DICOM format (Figure 4.2). The femur reconstruction process is built upon voxel sizes and a number of metadata information including intercept slope, rescales slopes, image position of the patient, location of the femur in the slice, DICOM slice thickness, and some other relevant information, which all are obtained from the DICOM files of the segmented QCT slices, following the sequential order of the instance numbers of the QCT slices. Furthermore, to calculate the intercept and slope, we stack the pixel arrays of the scans using the instance numbers. We convert the pixel values to HU values, and calculate the intercept and slope of the segmented femur, considering the HU values of air as the reference.

4.4.1 Stacking of Slices

The slices have been resampled according to the spacing obtained from the slices earlier. The newly segmented DICOM slices are then stacked on top of each other based on the new spacing

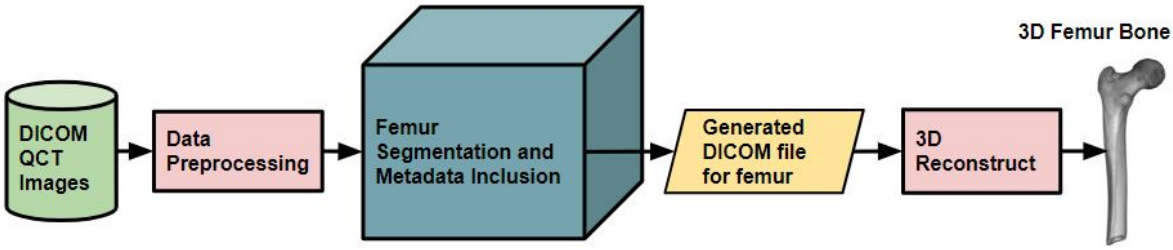


Figure 4.2: Pipeline for 3D femur reconstruction.

values. For 3D femur reconstruction, we use cubic spline interpolation, which is commonly used to reconstruct a continuous function or curve from discrete data points [68]. During the 3D reconstruction from 2D images, spline interpolation can be used to estimate the voxel values in between the actual image slices. To obtain the cubic spline interpolation function $f(x, y, z)$ (Equation 4.9), a set of input voxel data points, v_i at (x_i, y_i, z_i) from the segmented QCT slices have been considered.

$$f(x, y, z) = \sum_{i=0}^{n} N_i(x, y, z) \cdot v_i \tag{4.9}$$

where n is the number of input data points, and $N_i(x, y, z)$ are the cubic B-spline polynomial function, which is a continuous function and has continuous first and second-order derivatives. The cubic spline interpolation method is used to estimate the voxel values at any point in between the actual image slices, allowing for smoother 3D reconstruction of the image data. The coefficients of the cubic polynomials are determined by imposing these conditions on the polynomials. The cubic polynomial functions are defined recursively as shown in Equation 4.10 and Equation 4.11.

$$N_i^1(x) = \begin{cases} 1, & \text{if } x_i \leq x \leq x_{i+1}. \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

$$N_i^k(x) = \frac{x - x_i}{x_{i+k-1} - x_i} N_i^{k-1}(x) + \frac{x_{i+k} - x_i}{x_{i+k} - x_{i+1}} N_{i+1}^{k-1}(x) \tag{4.11}$$

where k is the degree of the spline, and in this case, $k = 3$. The cubic spline interpolation method provides smooth and continuous interpolation of the data, which is important for accurate 3D reconstruction of the femur from the segmented DICOM slices. The cubic spline interpolation method provides a smooth and continuous interpolation of data, which is important for an accurate 3D reconstruction of a femur from the segmented DICOM slices.

4.4.2 Volumetric Reconstruction

For volumetric reconstruction, we have utilized Python "Vedo" libraries [69], which is primarily designed for 3D visualization and image processing built on top of VTK (Visualization Toolkit) library functions [70]. In particular, the volume class from the Vedo package has been used for the 3D reconstruction from 2D images. The 3D volume regeneration process involves stacking of the 2D segmented DICOM images along the z-axis based on the origin, re-sampled voxel spacings, and original dimensions. We also have utilized static cast functions to convert the void pointer to correct the scalar type. This static-cast function converts the pixel array to float64 format which is the VTK volume mapper format for VTK volume rendering. Finally, a VTK volume mapper has been also used for rendering.

In Algorithm 6, we provide the pseudo code for our 3D volumetric femur reconstruction method from QCT images. The algorithm begins by extracting the slice thickness of the QCT slices from the updated metadata. This is achieved by utilizing either the slice location or image position of the patient. Next, we extract the Rescale Slope and Rescale Intercept values from the QCT images. These values are essential for calculating the position of each pixel in the three-dimensional space accurately. Additionally, we retrieve the voxel spacing information from the QCT slices. This information, combined with the pixel spacing, is used to perform B-Spline Interpolation, resulting in a smooth and continuous 3D volumetric reconstruction of the femur. Following this step, we normalize the volume object between 0 and 255. This standard normalization process prepares the volume for further processing, such as creating an STL file, which is our output format for femur segmentation and 3D reconstruction.

4.4.3 Surface Mesh Generation

To generate the 3D surface mesh of the femur, the "isosurface" function of the Vedo libraries has been employed. The generated 3D isosurface represents the boundary of the material of interest in the volume. To create a 3D mesh that can be rendered and manipulated, we triangulate the isosurface by subdividing the surface area into triangles connecting the vertices of the surface. The VTKPolyData, a 3D data structure, has been used in this process, and the VTK Marching Cubes algorithm [71] has been employed to convert the resulting VTKPolyData into a NumPy array of float64 to generate STL format—a suitable 3D mesh format.

In Algorithm 7, we present the pseudo code for surface mesh generation using the isosurface function. The algorithm takes the 3D volumetric object as input and produces the 3D femur in STL format as output. Within the algorithm, we initially read the volume object using the "VTKReader" function. This function allows us to access the volumetric data representation. Next, we pass the input object to the "DiscreteMarchingCubes" function, which implements the VTK Marching Cubes algorithm developed by Lorensen *et al.* [71]. This algorithm generates the isosurface of the femur from its 3D volumetric representation. The resulting isosurface is then

Algorithm 6 3D volumetric reconstruction of proximal femur by spline interpolation.

Require: QCT images in DICOM format
Ensure: 3D femur reconstruction, Volumetric data

$StoreFile \leftarrow FolderPath$
$Total\_QCT \leftarrow length(StoreFile)$
while $QCT < Total\_QCT$ **do**
  $DS \leftarrow Read\_DICOM(QCT)$
  $Img \leftarrow DS[PixelArray]$
  $QCT[InstanceNo] \leftarrow DS[InstanceNo]$
  $Sort(QCT) \leftarrow QCT[InstanceNo]$
  $QCT[PixelSpace] \leftarrow DS[PixelSpace]$
  $Slice\_Thickness \leftarrow |DS_{current}[ImagePositionPatient] - DS_{next}[ImagePositionPatient]|$
  if $Slice\_Thickness = 0$ **then**
    $Slice\_Thickness \leftarrow |DS_{current}[SliceLocation] - DS_{next}[SliceLocation]$
  end if
  $DS[Slice\_Thickness] \leftarrow Slice\_Thickness$
  $QCT \leftarrow QCT + 1$
end while
$Img \leftarrow Int16(Img)$
if $Img = 1000$ **then**
  $Img[Img = 1000] \leftarrow 0$
end if
if $DS[RescaleIntercept]$ **then**
  $QCT[RescaleSlope] \leftarrow DS[RescaleSlope]$
else
  $QCT[RescaleSlope] \leftarrow 0$
end if
if $DS[RescaleIntercept]$ **then**
  $QCT[RescaleIntercept] \leftarrow DS[RescaleIntercept]$
else
  $QCT[RescaleIntercept] \leftarrow 1$
end if
if $QCT[RescaleIntercept] \neq 1$ **then**
  $Img \leftarrow QCT[RescaleIntercept] * Float64(Img)$
end if
$Img \leftarrow Int16(DS[RescaleIntercept]) + Int16(Img)$
$Spacing \leftarrow map(Slice\_Thickness, QCT[PixelSpace])$
$Img \leftarrow B - SplineInterpolation(Img, Spacing)$
$Img \leftarrow normalize(Img, 0, 255)$
$volImg \leftarrow Volume(Img)$
$volImg \leftarrow SmoothMedian(volImg, neighbours(3))$

passed to the "VTKSTLWriter" function to create the STL file of 3D femur.

Algorithm 7 3D surface mesh generation.

Require: 3D Volumetric femur object ($volImg$)
Ensure: 3D femur in STL format
1: $Img \leftarrow VTK\_VolReader(volImg)$
2: $Img \leftarrow Update(Img)$
3: $MarchingCube \leftarrow VTK\_DiscreteMarchingCubes()$
4: $MarchingCube \leftarrow SetInputData(Img)$
5: $MarchingCube \leftarrow Update(MarchingCube)$
6: $ImgData \leftarrow GetOutput(MarchingCube)$
7: $STL \leftarrow vtkSTLWriter(ImgData)$

4.5 Quality Enhancement and Noise Removal

The final step of the 3D reconstruction process focuses on the quality improvement of the generated volume by eliminating unwanted voxels, if there is any. We have provided the pseudo code for this stage in Algorithm 8. We have utilized the "Islands Removal" algorithm [72] to eliminate any detached islands of voxels around the 3D femur bone. This algorithm has been applied along with a threshold value that can determine the size of the isolated "islands" of voxels and the filter only kept the largest connected island of voxels. To obtain an optimal 3D femur surface while maintaining the femur geometry, manually optimized threshold values, ranging from 0.75 to 45.25 along the three axes have been used to eliminate isolated noise above the surface for each patient. The quality of the 3D surface was further enhanced by applying a "Median Smoothening" function to close any gaps by taking the median voxel values from the neighboring voxels. This process also has helped to eliminate additional noise in the 3D femur surface. Finally, we convert the reconstructed 3D femur in STL format to allow for further visualization and inspection. We have presented an example of the transformation of femur after noise removal in Figure 4.3.

In Algorithm 8, we observe the usage of the condition $TotalIsland > 1$ to obtain the largest connected island. This step aims to extract the primary connected region representing the 3D femur while eliminating other detached materials that may exist within the QCT images. Furthermore, we employ the median values from the closest three neighboring points to smooth the surface of the 3D femur bone. This smoothing process helps reduce noise and irregularities in the reconstructed surface. During the hole-filling procedure, we introduce a threshold value to limit the permissible hole sizes. This approach ensures that excessively large holes are not filled, as it could result in a deformed representation of the femur bone. By imposing a threshold on the hole-filling process, we strive to achieve an accurate 3D femur bone reconstruction with minimal distortions.

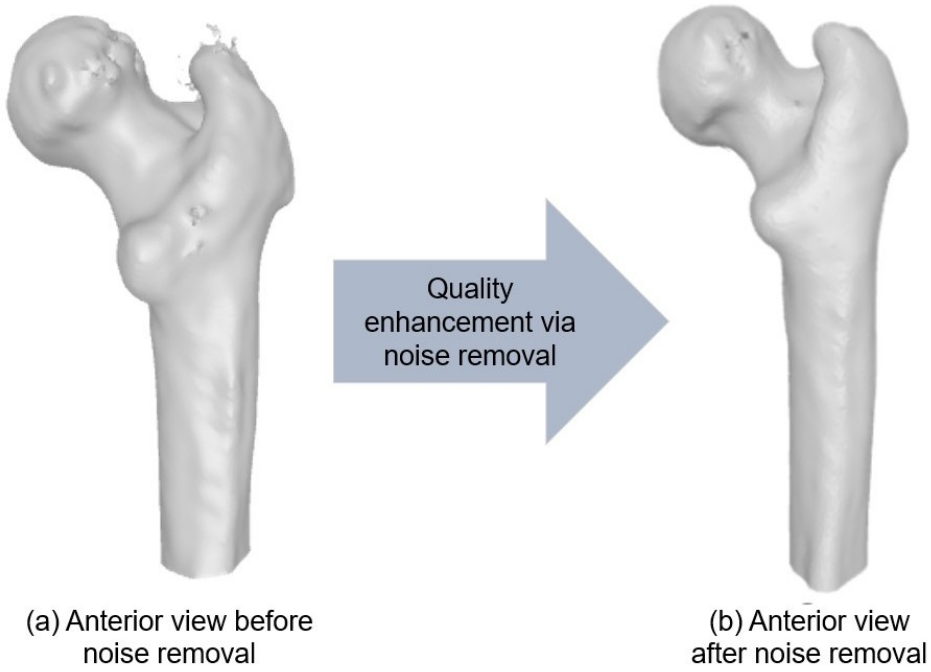


Figure 4.3: An example of quality improvement using "Islands Removal" algorithm and "Median Smoothing" algorithm.

Algorithm 8 Quality enhancement by noise removal.

Require: 3D isosurface of femur object in STL ($stlImg$)
Ensure: 3D femur in STL format

$Img \leftarrow VolIsosurface(stlImg)$
$TotalIsland \leftarrow IslandCount(Img)$
while $TotalIsland > 1$ **do**
  $Img \leftarrow Threshold_{patient\_specific}(Img)$
  $TotalIsland \leftarrow IslandCount(Img)$
end while
$Median_{x,y,z} \leftarrow GetMedianUptoNeighbour(3)$
$Img \leftarrow MedianSmooth(Img, Median)$
while $DetectHoles(Img)$ **do**
  $Img \leftarrow FillHoles(Size)$
  if $DetectHoles(Img) < Size(\frac{3}{4} * DetectHoles(Img))$ **then**
    $ExitLoop$
  end if
end while
$Img \leftarrow STL(Img)$

Chapter 5

Results and Analysis

The primary goal of this research is to achieve the patient-specific 3D reconstruction of the segmented proximal femur using an automated end-to-end SSDL model. To evaluate the performance of the model, a testing dataset consisting of 10 patients and a total of 20 femurs (including both left and right femora) were used. Among these 10 patients, four were already known about the training process of SSDL segmentation model through a validation process. In comparison, the remaining six patients were completely, new and independent, unseen to our segmentation model.

The outcomes of this work encompass two aspects: the 2D segmented Quantitative Computed Tomography (QCT) images of the femur and the 3D reconstructed femur. Therefore, the effectiveness of our model was assessed separately for both outcomes. The performance evaluation involved a comprehensive analysis in terms of both qualitative and quantitative measures. Furthermore, the evaluation was conducted on both the validated patients (included in the model's validation process) and the independent unseen patients to assess the robustness of our proposed SSDL model. Additionally, we have implemented MultiResUNet segmentation model [73]. The obtained results from these experiments are also presented and discussed.

5.1 Segmented Femur

In this section, we provide qualitative and quantitative results and analysis of femur segmentation.

5.1.1 Qualitative Results

The 2D segmented femur at different locations are shown and compared qualitatively (visually) with the ground truth binary mask in Figure 5.1. The visual comparison depicts that the contours of the segmented femur via our proposed SSDL method match very well with the ground truth binary mask. This qualitative comparison suggests that our model for segmentation is accurate in

isolating the regions of interest in the QCT slices. The proposed SSDL model is able to preserve the structural integrity of the femur's internal anatomy such that the hollow region of the femoral shaft is turned into solid with gradually changing bone density at the proximal part comprising trochanter and femoral head. The model is also able to capture the complex geometrical structure at the trochanteric regions (shown in Figure 5.1). Furthermore, the framework is capable of accurately predicting the voxel values, which are later used for 3D femur reconstruction.

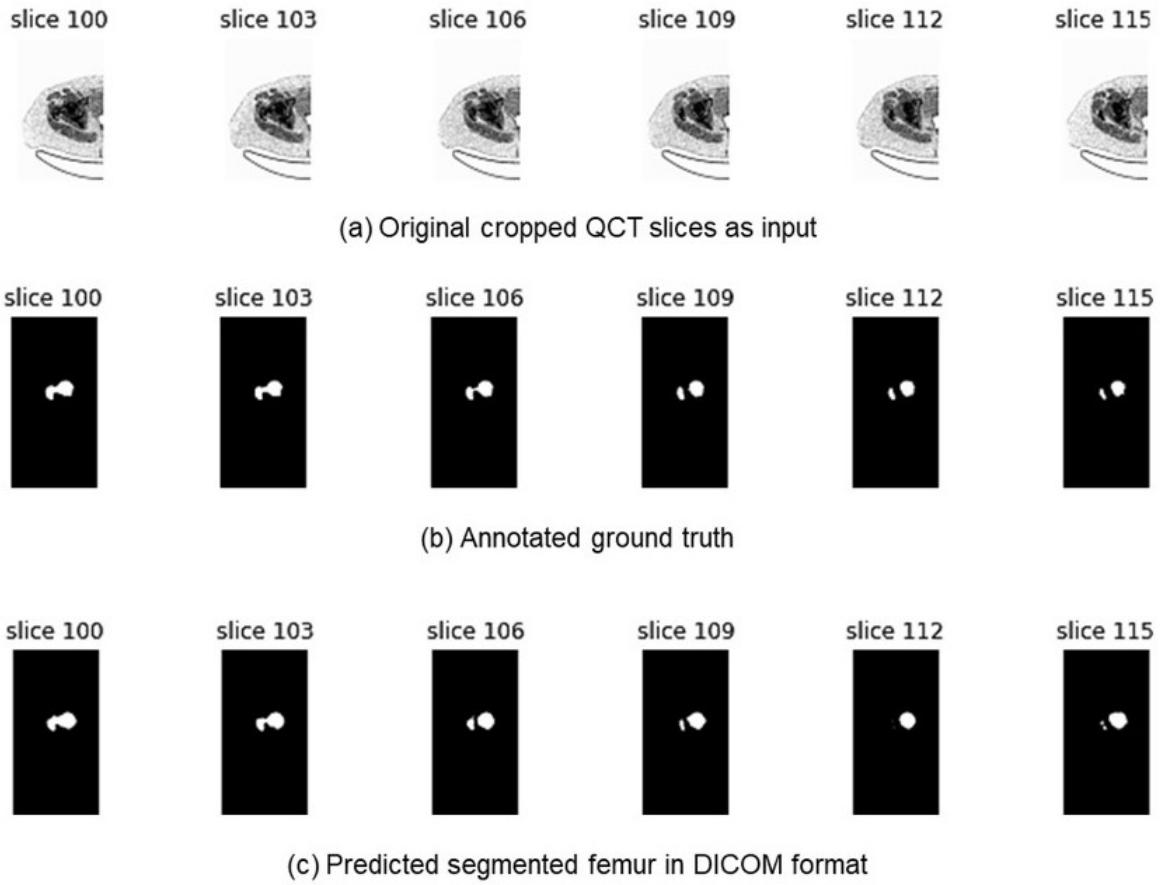


Figure 5.1: Segmentation of femur on different QCT images in DICOM format using SSDL segmentation model. (a) Input QCT slices, (b) Annotated ground truth, and (c) Predicted segmented femur.

5.1.2 Quantitative Results

Figure 5.2 shows a quantitative comparison between the area of the segmented femurs obtained from our SSDL model's segmentation unit and the ground truth. It is evident that our proposed framework is able to segment femur with high accuracy. The segmentation algorithm can accurately identify the boundaries of the femur in the images. The results indicate that the maximum difference between the area values obtained from automated segmentation and manual segmentation is relatively small, ranging from approximately 0.430 cm^2 to 2.574 cm^2. This

suggests that the segmentation algorithm can produce accurate and reliable measurements of femur area.

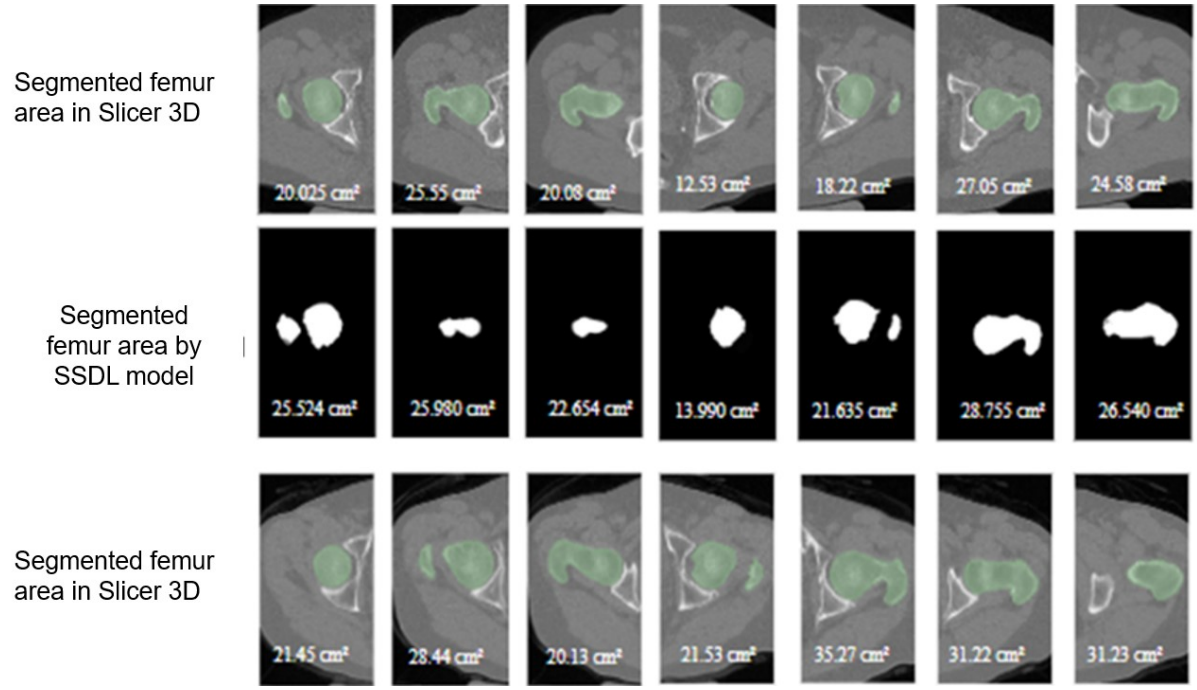


Figure 5.2: Comparison of femur area between (manually/semi-automatically) selected femur contour (green shaded area) on original DICOM image using 3D Slicer and fully automated segmented femur in DICOM format using proposed SSDL segmentation model. The QCT slices in the proximity of the hip joint are shown here as examples considering the most critical location in terms of segmentation.

A detail example is provided in Figure 5.3. In this figure, we observe some annotation errors highlighted in the green shaded area on the original DICOM images. These errors significantly contribute to the overall error in the surface area of the segmented femur. It is worth noting that the proximal head of the femur poses a challenge for both human annotators and our SSDL model. The presence of annotation errors in the green shaded area indicates that the manual annotation process is not entirely free from mistakes. These errors may arise due to the complex anatomical structures present in the proximal head region, which can make it difficult to accurately delineate the boundaries. Additionally, the automated SSDL model may encounter challenges in accurately segmenting this particular area, as it shares similarities with adjacent structures or exhibits variations in bone density. Understanding and addressing these difficulties are essential for improving the accuracy and reliability of the segmentation results, particularly in challenging regions like the proximal head of the femur. Further research and refinement of the segmentation algorithms and annotation procedures can help overcome these challenges and enhance the performance of the automated femur segmentation system.

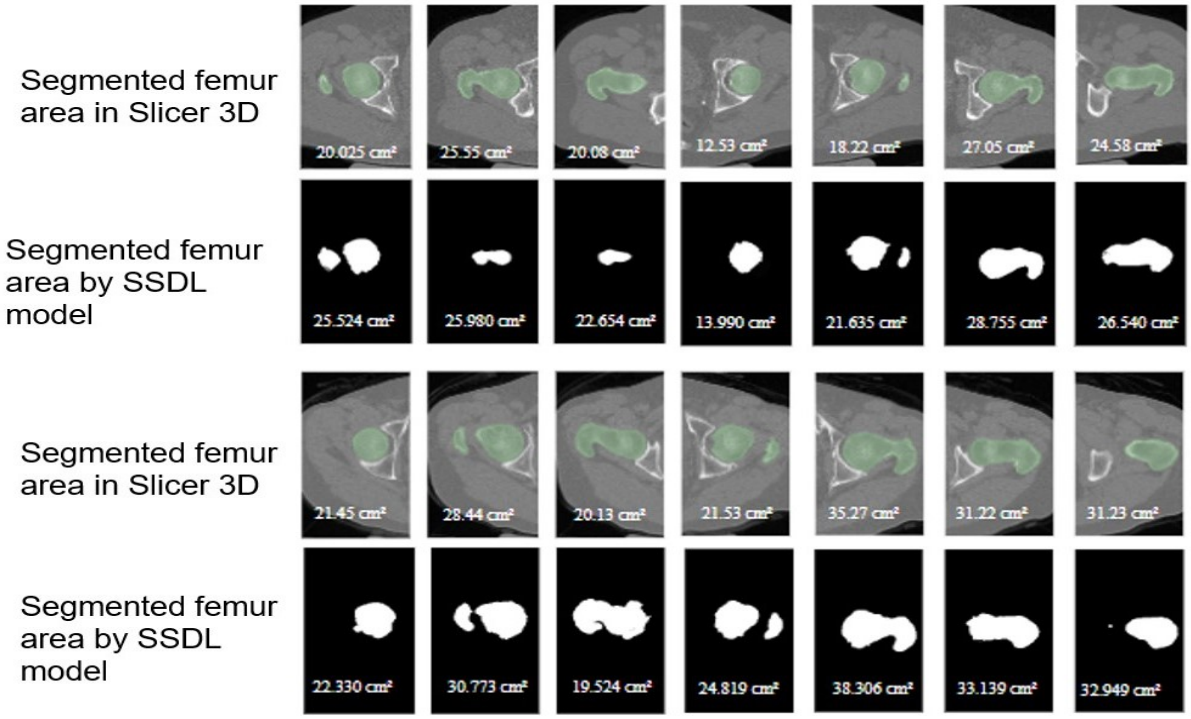


Figure 5.3: Comparison of femur area between (manually/semi-automatically) selected femur contour (green shaded area) on original DICOM image using 3D Slicer and fully automated segmented femur in DICOM format using proposed SSDL segmentation model.

Training and Validation Performance

Figure 5.4 and Figure 5.5 showed the training and validation performance of the SSDL model. These plots illustrate the training and validation processes for both low-performing patients' data and high-performing patients' data, respectively.

Analyzing the figures, we can observe distinct differences in the validation processing between the low-performing and high-performing patient data. In the case of the low-performing patients' data, the validation process appears to be erratic and rocky, with fluctuations in performance. This indicates that the SSDL model struggles to generalize well to this particular set of data, leading to inconsistent validation results. On the other hand, the high-performing patients' data exhibit smoother slopes in both the training and validation curves. This suggests that the SSDL model is able to effectively learn and generalize from this data, resulting in more stable and consistent performance. Interestingly, in the case of the high-performing patients' data, we can observe that the training and validation slopes gradually merge together as the training progresses. This convergence indicates that the model is effectively learning from the training data and achieving the highest possible performance. This alignment between the training and validation curves further highlights the robustness and effectiveness of the SSDL model in accurately segmenting the femur for high-performing patients. These findings demonstrate the importance of evaluating the performance of the SSDL model on different patient datasets, as it helps identify variations

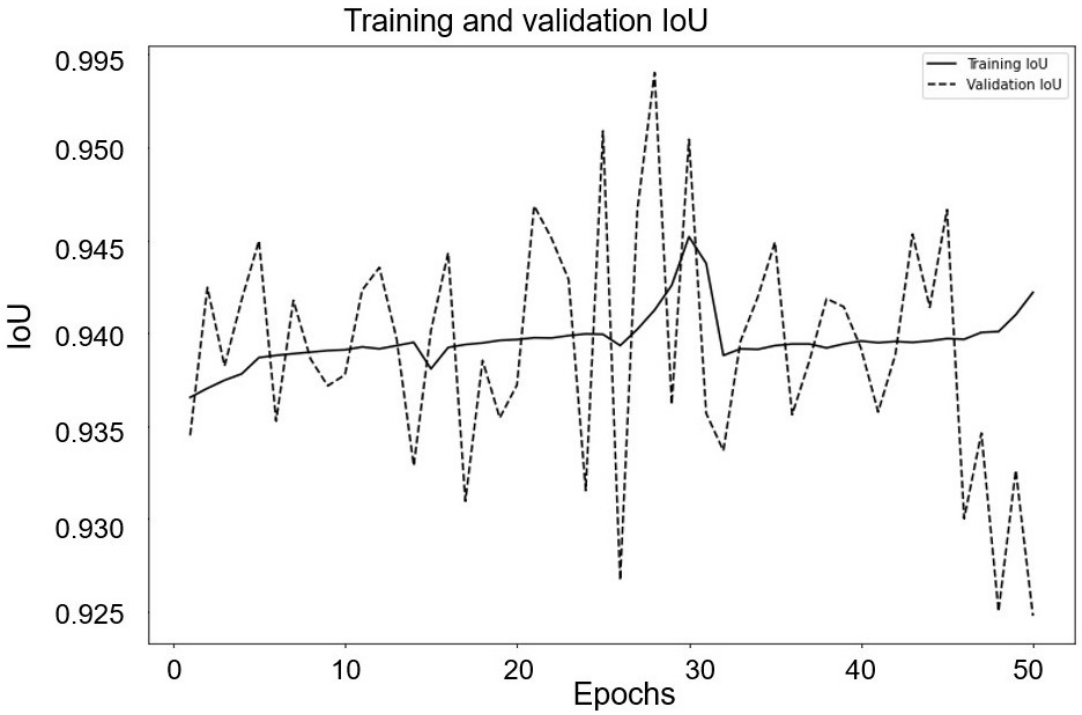


Figure 5.4: Line plot for mean IoU for training and validation data for the lowest performing patient.

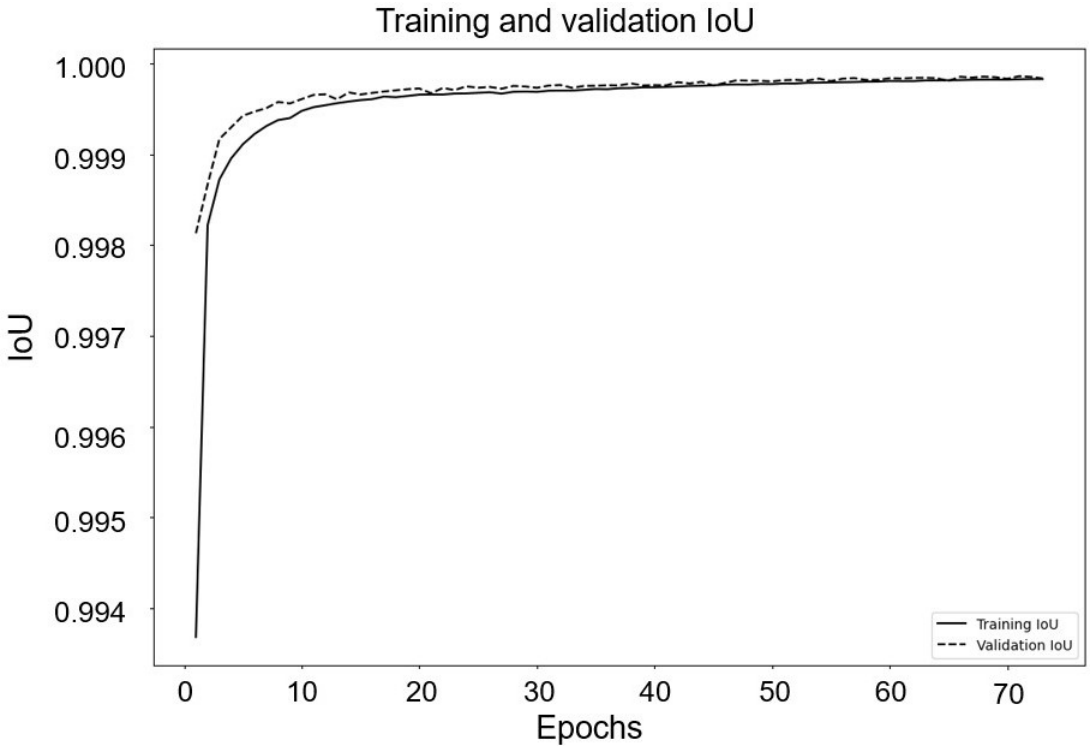


Figure 5.5: Line plot for mean IoU for training and validation data for the highest performing patient.

in performance and provides insights into the model's ability to handle different types of cases. Furthermore, the smooth slopes observed in the high-performing patients' data highlight the potential for achieving optimal performance through effective training and validation strategies.

The same trend can also be observed in the training and validation process in DSC values for both the high-performing patients' data (Figure 5.6) and the low-performing patients' data (Figure 5.7). Here, the slopes for the training and validation process never merge and always stay almost parallel for the lowest-performing patient. This further strengthens the findings and adds more credibility to our results.

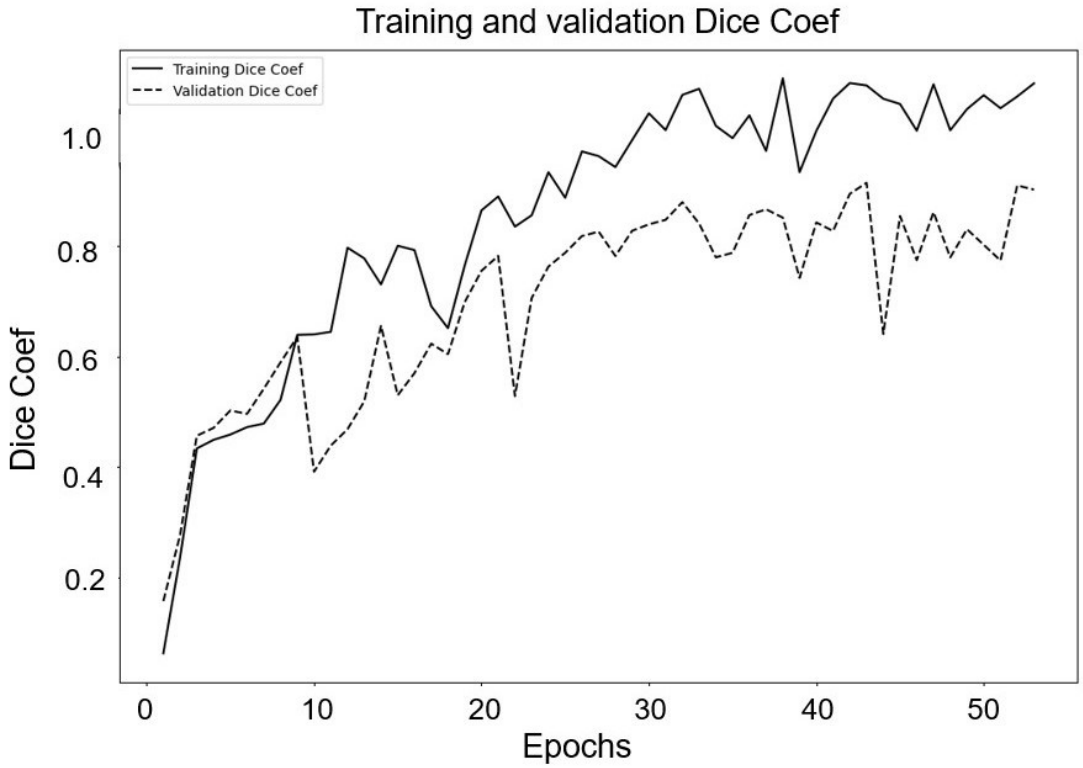


Figure 5.6: Line plot for DSC for training and validation data for the highest performing patient.

Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) are valuable tools for evaluating and comparing the performance of binary classification models, providing insights into their discriminatory power, robustness to imbalanced data, and threshold selection. Figure 5.8 presents the ROC curve and the corresponding AUC value, which are used to assess the performance of our SSDL model in femur segmentation from QCT images. The ROC curve illustrates the trade-off between the true positive rate and the false positive rate at various classification thresholds. The AUC value quantifies the overall discriminative power of the model, with higher values indicating better performance.

In our study, the AUC value obtained is 92.4%, which demonstrates that our SSDL model achieves a high level of accuracy and reliability in segmenting femurs from QCT images. This indicates that the model effectively distinguishes between positive (femur) and negative (non-

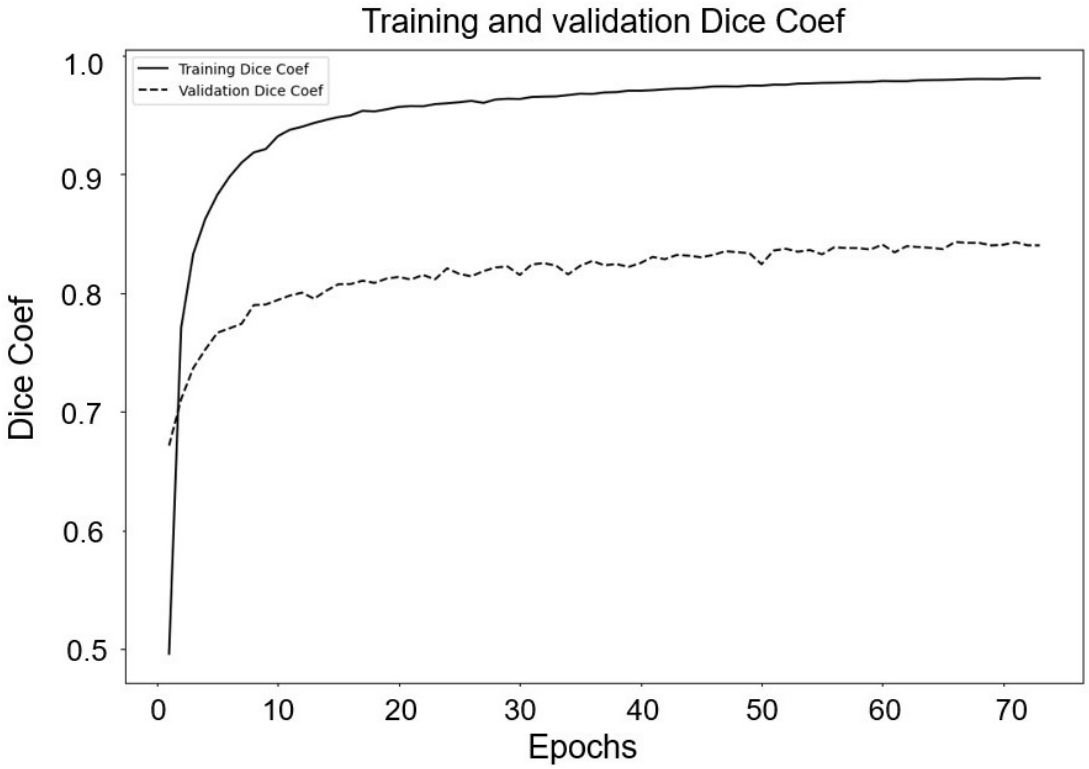


Figure 5.7: Line plot for DSC for training and validation data for the lowest performing patient.

femur) instances, with a minimal rate of false positives and false negatives. With the high AUC value of 92.4%, our SSDL model is successful in precise segmentation of the femur.

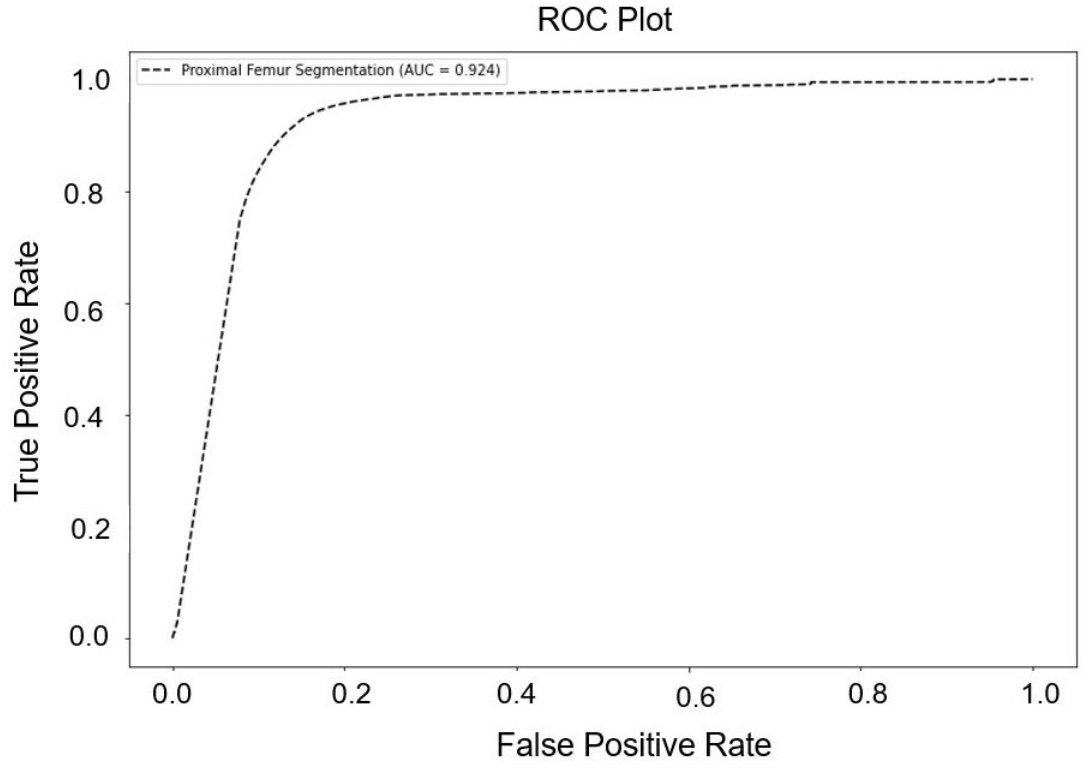


Figure 5.8: Plot of ROC curve of SSDL model's performance. AUC = 92.4%

Cross Validation

Cross-validation is a vital step of any ML/DL experiment, and an appropriate data split can significantly impact the model's performance. In this study, we employed the LOOCV method to split our dataset into training and testing sets. This approach involves dividing the dataset into multiple subsets, with each subset consisting of one sample as the testing set and the rest as the training set. During the experiments, an 8-fold cross-validation was used, where the data was divided into eight subsets of patients, and each subset served as a testing set once for one patient. In comparison, the remaining seven subsets form the training set. By leveraging the LOOCV technique, we ensure that our model is trained and evaluated on a diverse set of data and the results are robust to variations in the data distribution.

Performance Summary

Table 5.1 shows the segmentation performance of our proposed SSDL model quantitatively for both validated and unseen patients' data. All of the performance indicators exhibited accuracy over 90% even with such a limited dataset. As we can see, our SSDL segmentation model had achieved DSC, average precision, and sensitivity of 0.918 ± 0.025, 0.996 ± 0.003, and 0.995 ± 0.004 for the unseen patients' dataset while for the same performance metrics, our model had achieved 0.992 ± 0.008, 0.996 ± 0.003, 0.995 ± 0.004 for validated patients' data. From these performance indicators, we observed the performance difference between the validated and unseen patients is nearly 7%. The slight difference of the performance between the validated and unseen dataset may have originated from the variation of bone density distributions, which are likely affected by osteoporosis, bone necrosis, etc. We do not have prior knowledge of any disease for the patients in the considered dataset. Our SSDL model was hyper-tuned using validation data. We find that mean IoU and specificity showed good performance for both validation and unseen patients. Hence, we can claim that with high specificity, our model can detect the pixels for non-femur values with high correctness even though a slight variation is observed for DSC, Average Precision, and Sensitivity. However, it is evident that our SSDL model is capable of segmenting femurs from DICOM images of unseen patients with comparable accuracy.

In addition, the performance metrics in Table 5.1 also suggests that our automated model is generic, it can perform any femur segmentation, and this model can be applied to any patient with geo-diversity like patients with different races, ethnicity, and age groups etc. By examining Table 5.1, it is evident that our SSDL model demonstrates superior quantitative performance when utilizing the PReLU activation function. We theorized that this occurance happened due to some negative pixels denoting some proximal femur regions. ReLU activation function works by discarding all negative input values, effectively setting them to zero. However, in the case of the proximal femur where negative pixels may exist, this can lead to a loss of important information.

Table 5.1: Quantitative performance evaluation of femur segmentation by SSDL model.

Activation Function	Patient's Status	DSC	mean IoU	Average Precision	Sensitivity	Specificity
ReLU/ PReLU	Validated	0.992 ± 0.008	0.995 ± 0.005	0.996 ± 0.003	0.995 ± 0.004	0.998 ± 0.002
ReLU	Unseen	0.918 ± 0.025	0.983 ± 0.016	0.923 ± 0.024	0.927 ± 0.02	0.999 ± 0.002
PReLU	Unseen	0.937 ± 0.025	0.953 ± 0.018	0.949 ± 0.023	0.997 ± 0.002	0.999 ± 0.002

In contrast, the PReLU activation function addresses this issue by introducing a parametric value, denoted as α, to handle the negative cases. This allows the negative pixels of the femur to be preserved, preventing the loss of valuable information during the activation process. Therefore, by employing the PReLU activation function, our SSDL model is able to retain and utilize the negative pixel values present in the proximal femur areas, resulting in improved segmentation performance. This adaptive activation function enables the model to capture and leverage the full range of pixel intensities, contributing to the enhanced quantitative outcomes observed in Table 5.1.

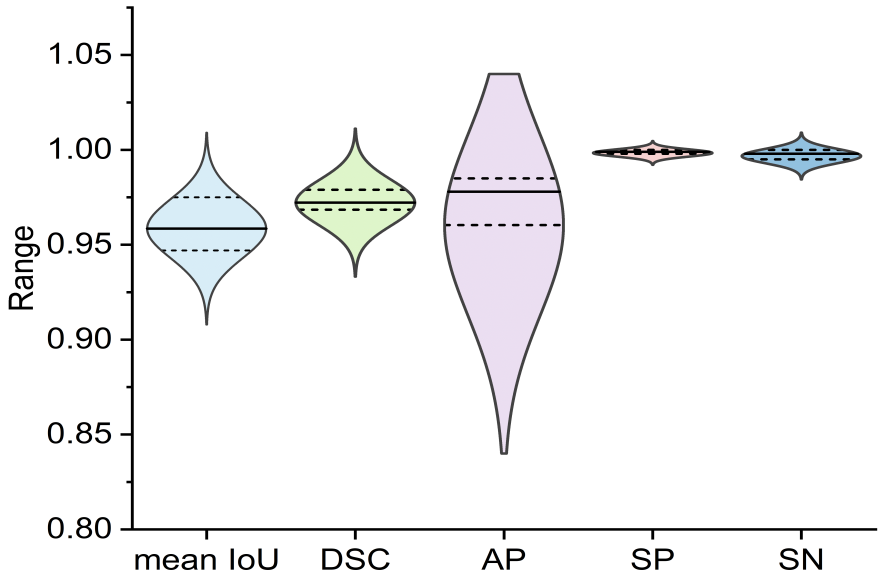


Figure 5.9: Violin plots showing the distributions of performance indicator values for the segmentation results of femurs in testing dataset using SSDL model. The solid line represents median; dotted lines are for 1st (bottom) and 3rd (top) quartiles. Here, AP: Average Precision, SP: Specificity, SN: Sensitivity.

Here, we have shown the distributions of the performance indicators (DSC, mean IoU, Average

Precision, Sensitivity, and Specificity) as a continuous approximation of the probability density function (PDF), computed using kernel density estimation (KDE) in violin plots in Figure 5.9. The wider regions of the density plots indicate the performance indicator values occur more frequently. The narrower sections represent a lower probability, indicating performance indicator values that occur less frequently. A key advantage that PDFs have over histograms is that the use of a continuous function avoids the issue of having to choose bins. Hence, this gives a more accurate shape of the distribution independent of the number of bins. The violin plot uses KDE to compute an empirical distribution of training data, and therefore, it better reveals the information contained in the training and more convincingly suggests multimodality, if there is any. It is apparent that the distributions of all performance metrics are normal, indicating mean and median are nearly the same.

From the outcomes, we can clearly observe the potential of our automated U-Net mask generator model of SSDL for reconstructing the 3D femur. Although the preliminary model is trained with an extremely small number of data points, it shows its ability to generate a mask image from the CT scans input to isolate the femur from the other hip bones. Furthermore, by creating different mask images to segment the femur from the CT scan images, we can conserve the image data without any modification. To the best of our knowledge, the automated construction of femur using a DL framework is yet to be achieved successfully. It must be mentioned that it is considerably difficult and laborious task to annotate each DICOM file for supervised learning. Thus, we decided to use a semi-supervised DL model trained with a combination of annotated and raw DICOM slices. Most importantly, our segmentation framework works without any image augmentation, preserving all the metadata associated with original DICOM images. The results are promising and demonstrate its potential for accuracy improvement by increasing the training data. Accuracy can be further enhanced by resampling segmentation, cropping, padding, smoothing, etc., if necessary [37, 45].

5.1.3 Results from MultiResUnet

Figure 5.10 depicts the performance of the adopted semi-supervised MultiResUNet model in femur segmentation from different QCT images. The figure illustrates that the model excels in accurately segmenting the geometrically complex parts of the femur, where a higher number of annotations are available for training (Figure 5.10 (b)). These geometrically complex regions can be captured by the model, resulting in precise segmentation. However, the performance of the MultiResUNet model diminishes when it comes to segments of the femur with less geometric complexity. These regions, which have fewer annotations available for training, pose a challenge for the model. The model struggles to accurately predict the boundaries and structures in these areas, leading to suboptimal segmentation results (shown in Figure 5.10(b)).

This observation suggests that the performance of the MultiResUNet model is heavily dependent

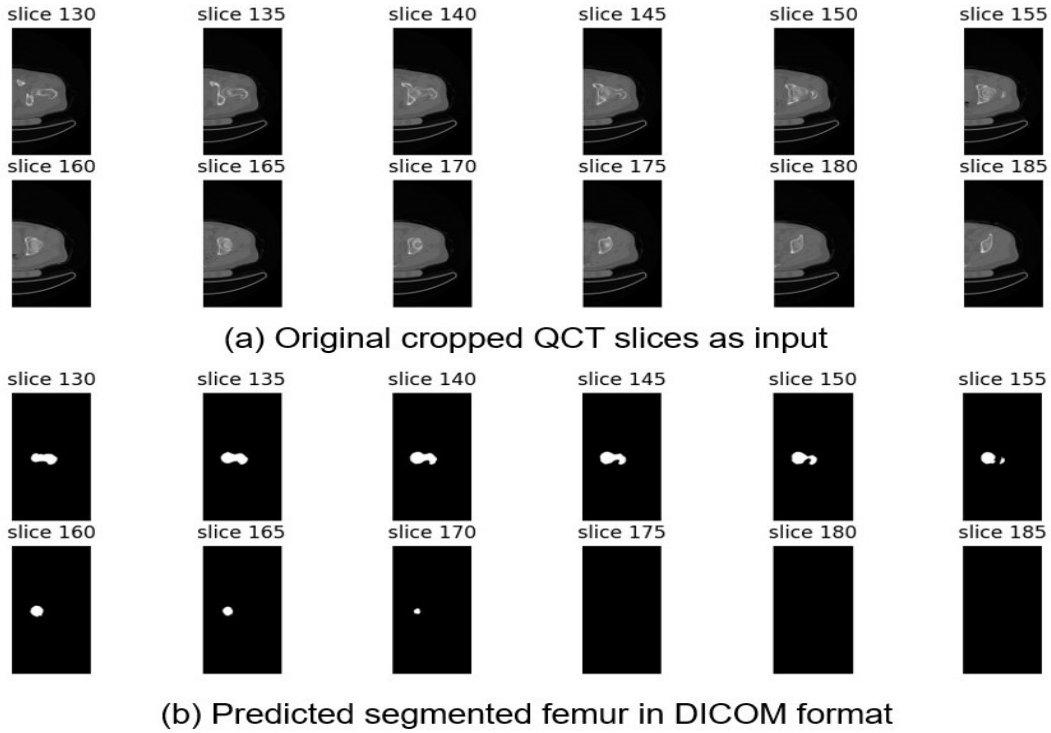


Figure 5.10: Segmentation of femur on different QCT images in DICOM format using MultiResUNet model. (a) Input QCT images and (b) Predicted segmented femur.

on the amount and quality of training data available for different parts of the femur. When provided with sufficient annotations for the complex regions, the model exhibits better segmentation capabilities. Conversely, for regions with limited training annotations, the model's performance deteriorates. These findings highlight the importance of adequate training data and annotation coverage for achieving optimal performance with the MultiResUNet model. Further efforts to improve the model's performance in less geometrically complex femur structures can involve augmenting the training dataset with additional annotations and exploring alternative training strategies to enhance its ability to generalize to diverse anatomical variations.

Table 5.2 provides a quantitative assessment of the segmentation performance of the adopted semi-supervised MultiResUNet model, both for validated and unseen patients' data. The table highlights various evaluation metrics, such as the DSC, mean IoU, Average Precision, Sensitivity and Specificity, which serve as indicators of the model's accuracy and precision in femur segmentation.

Table 5.2: Quantitative performance evaluation of femur segmentation via MultiResUNet model.

Patient's Status	DSC	mean IoU	Average Precision	Sensitivity	Specificity
Validated	0.892 ± 0.08	0.999 ± 0.0001	0.961 ± 0.03	0.955 ± 0.007	1.0 ± 0.00
Unseen	0.845 ± 0.03	0.999 ± 0.0	0.906 ± 0.006	0.921 ± 0.03	0.999 ± 0.0001

From the presented results, it has been found that our proposed model SSDL performs better than the MultiResUNet model in terms of quantitative metrics. Specifically, the DSC and Average Precision values are lower for the MultiResUNet model. This discrepancy in performance can be attributed to the model's challenges in accurately segmenting the femur, particularly in regions with less geometric complexity and in hard tissues within the shaft region. These findings underscore the importance of choosing an appropriate segmentation model for femur segmentation tasks.

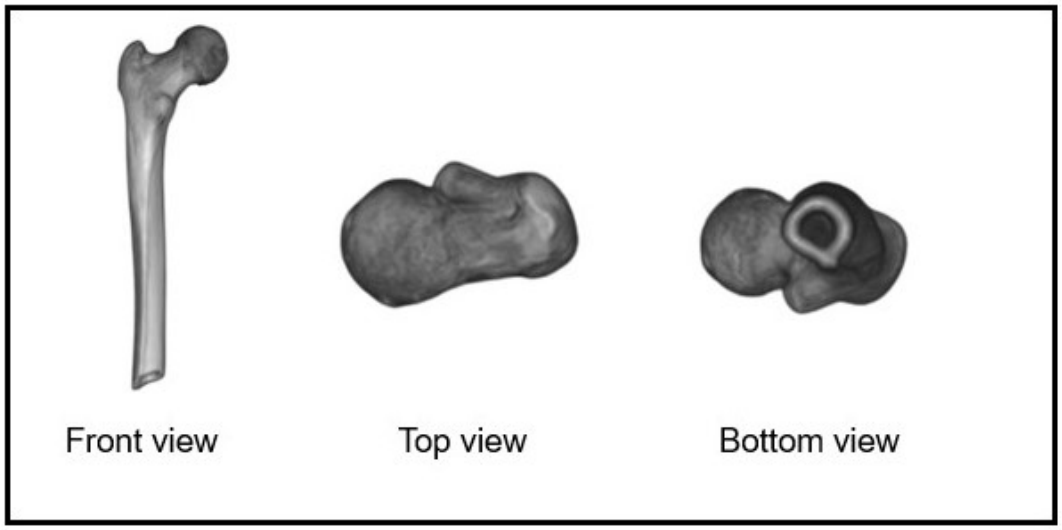
5.2 Reconstructed 3D Femur

The final goal of this study was to generate a 3D proximal femur by our adopted SSDL model without manual intervention. We now provide both qualitative and quantitative results and analyses of femur reconstruction.

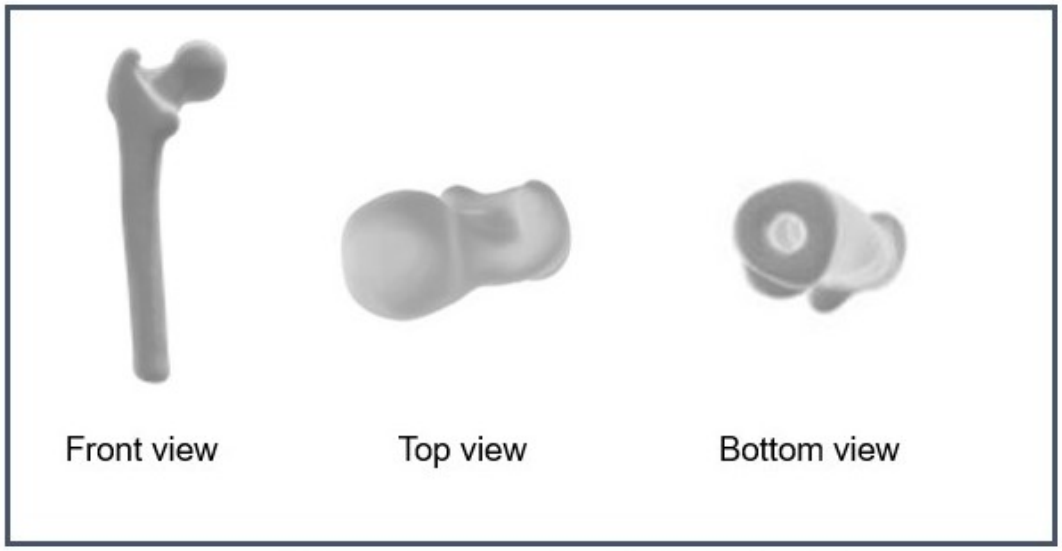
5.2.1 Qualitative Results

In this section, we provide qualitative results and analysis. Figure 5.11(a) and Figure 5.11(b) show different views on coronal (Front View) and transverse planes (Top and Bottom View) of a 3D reconstructed femur by SSDL model and 3D Slicer, respectively. It is to be noted that our model successfully segmented all the unwanted components of the QCT images, including muscle, fat, especially bone marrow, which is reflected as the hollow shaft section as shown in Figure 5.11(a) (Bottom View). To the best of our knowledge, no prior works considering different ML/DL models were able to generate a hollow shaft [5, 39] that gradually vanishes toward the proximal end and turns into the solid mimicking trabecular bone at the proximal end. The bone models were also compared qualitatively with manually reconstructed femur using the widely used and publicly available software, *3D Slicer* (https://www.slicer.org/). From the comparison with manually reconstructed 3D femur, we observe noticeable thickness differences in the internal hollow section. Later, this visual discrepancy was also reflected in the quantitative results for the reconstructed 3D femur.

Both qualitative and quantitative comparisons were conducted for the 3D reconstructed femur bone converted into .stl format (Figure 5.12 and Figure 5.13). The comparative analysis has been done based on the two separated datasets—validated and unseen patients' dataset as mentioned earlier. Figure 5.12 shows the anterior view of 3D reconstructed left and right femurs of a validated patient (Figure 5.12(a)) and an unseen patient (Figure 5.12(b)). Figure 5.13 shows a greater detail by representing both anterior and posterior views of the right and left femurs of two unseen patients. It is apparent that our framework can produce a smoother surface on the anterior side than on the posterior side of the femur.



(a) Automated reconstruction by SSDL



(b) Manually reconstructed by Slicer 3D

Figure 5.11: An example of a 3D reconstructed proximal femur with our proposed SSDL model (a) and 3D Slicer (ground truth) (b). Both figures show Posterior (Front) view on coronal plane, Top view, and Bottom view on transverse plane.

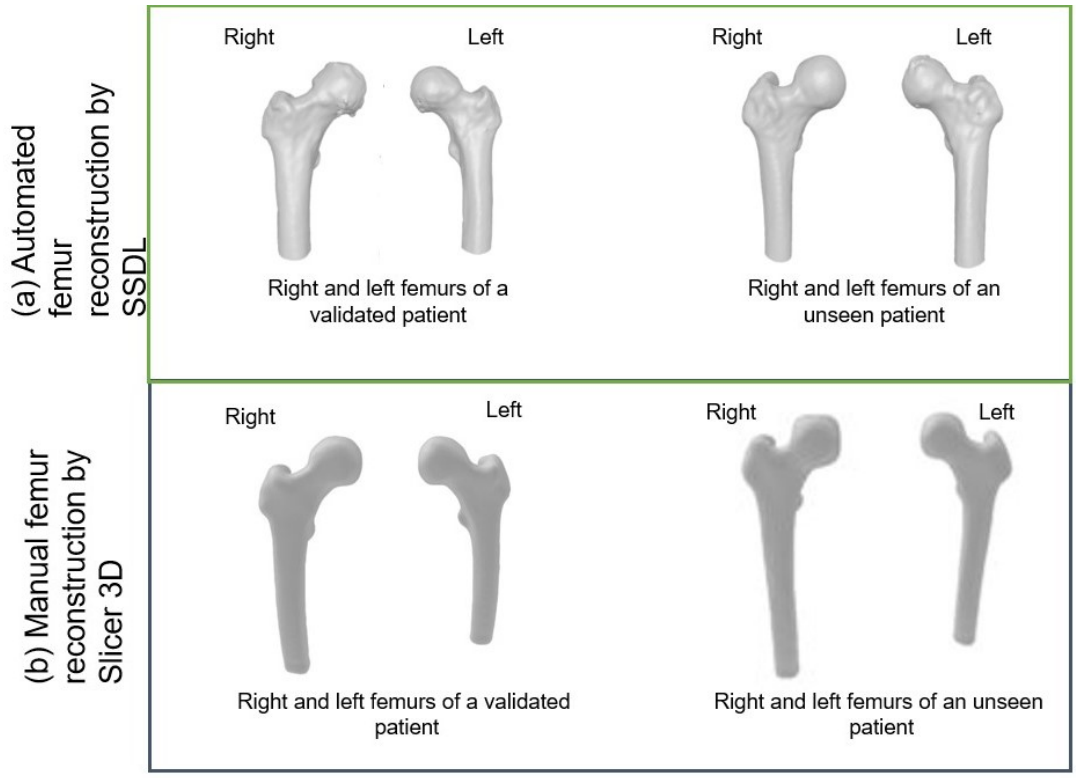


Figure 5.12: Examples of 3D femur reconstruction from segmented DICOM files via SSDL model and 3D Slicer (ground truth). Anterior views of right and left femurs of a validated patient (a) and an unseen patient (b).

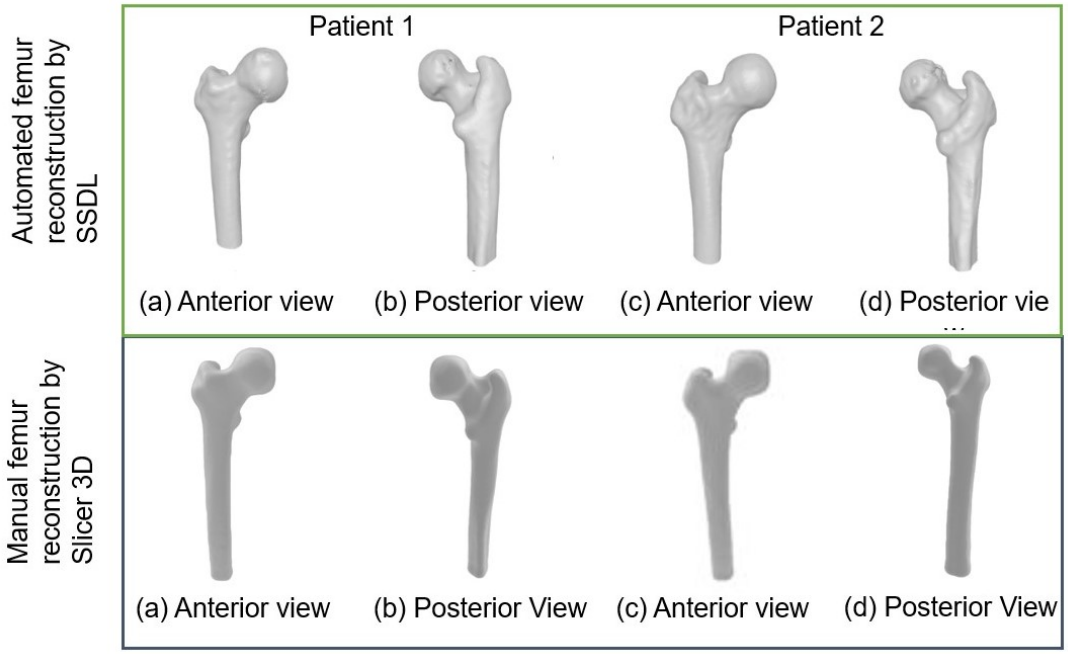


Figure 5.13: Examples of 3D reconstructed femurs (right only) of unseen patients (1 and 2) with anterior views (a and c) and posterior views (b and d) by SSDL model and 3D Slicer (ground truth).

5.2.2 Quantitative Results

The 3D reconstructed femurs by SSDL model have been quantitatively compared with manually reconstructed 3D femurs using *3D Slicer* based on surface area and volume. The differences in surface area and volume between the automated and manually generated 3D proximal femurs have been found as $9.958 \pm 6.610\ cm^2$ and $14.101 \pm 12.082\ cm^3$, respectively. We observe that the quantitative variations are fair enough because our dataset was relatively small dataset for training and testing. However, the deviations might be caused by the "smoothing" operation, which was rigorously used in manual intervention as well as due to the hollow section and its transition from the femoral shaft to the greater trochanteric region. Figure 5.14 shows the violin plots of the distributions of differences of volume and surface area between the femurs generated by SSDL and manually by *3D Slicer*. We find that both plots exhibit normal distribution and a wider distribution near the median with nearly equal mean and median values, which matches very well.

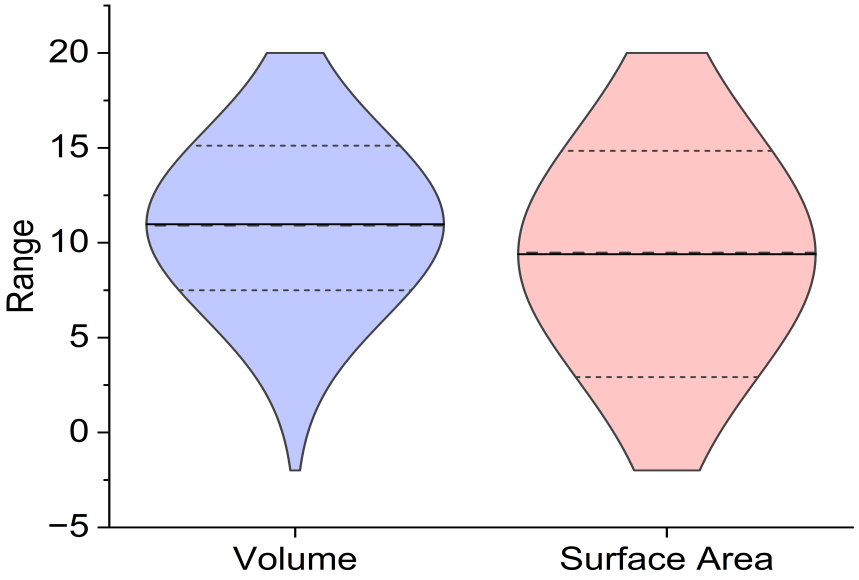


Figure 5.14: Violin plots showing the distributions of the differences in the volume of 3D reconstructed femurs and the surface area of segmented femur obtained via SSDL approach and manually.

Figure 5.15 illustrates the volume error, expressed as a percentage, which provides insight into the disparity between the ground truth and the predicted 3D femur volumes. The plot allows us to examine the variations in volume estimation across different femur samples. Upon observing the plot, it becomes evident that the volume errors for the majority of the reconstructed 3D femurs fall within a similar range, indicating consistent performance. However, it is worth noting

that femur 4 stands out as an outlier, displaying a considerably larger volume error compared to the other femurs. The plot serves as a valuable tool in assessing the accuracy of the 3D femur reconstruction process. The relatively consistent volume errors for the majority of the femurs suggest that the reconstruction method produces reliable and accurate results. However, the notable discrepancy observed for Femur 4 indicates the presence of unique characteristics or challenges specific to that particular femur, leading to a significant deviation from the expected volume.

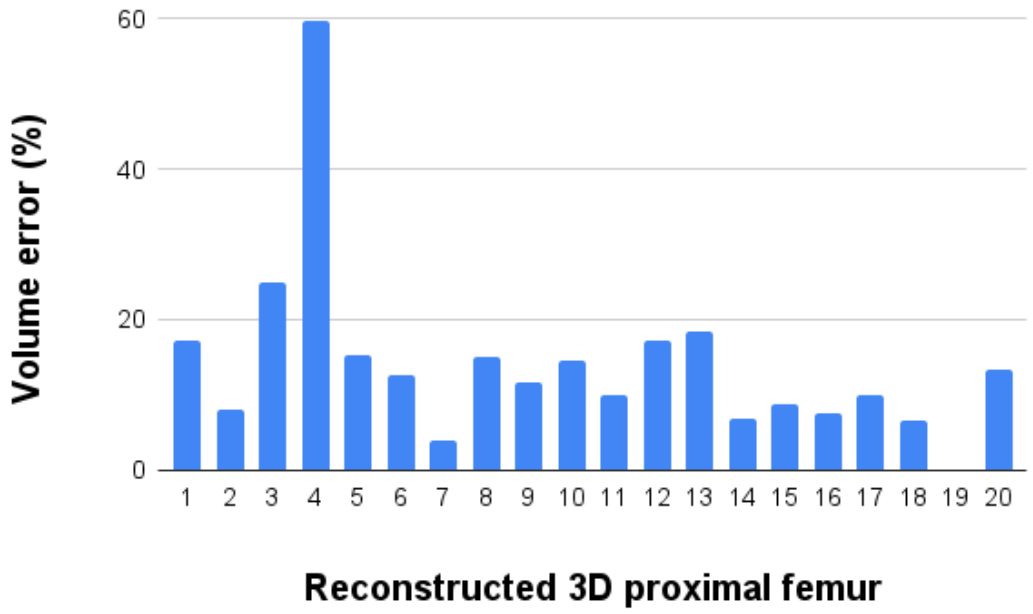


Figure 5.15: Volumetric error (%) between ground truth femur and predicted femur for each proximal femur (X axis represents each femur, Y axis represents error).

To further analyze the performance of our framework, we calculated several metrics, including *Root Mean Square Error* (RMSE), *Mean Average Error* (MAE), *Relative Error* (RE), R^2, and *Correlation Coefficient* values for both the surface area and volume of the 3D reconstructed femurs as presented in Table 5.3. The small values of RMSE, MAE, and RE indicate that our framework exhibits low error and better accuracy in generating 3D femurs. R^2 values are used to assess how well our framework fits the purpose of generating 3D femurs. Our 3D femur reconstruction framework shows low to moderate evaluation of variants with values close to 0.34 (for volume) and 0.6 (for surface area). However, with strong correlation coefficient values of 0.77 (for volume) and 0.85 (for surface area), our 3D femur reconstruction framework shows relatively small errors and a strong positive relationship between the evaluated variables, indicating that it performs well in the reconstruction of 3D femurs from our segmented DICOM files.

Table 5.3: Performance measure of 3D reconstructed femur by SSDL approach in comparison with manually reconstructed femur.

	RMSE	MAE	RE (%)	R^2	Correlation Coefficient
Volume (cm3)	33.820	29.556	6.610	0.338	0.771
Surface Area (cm2)	51.839	43.077	12.082	0.592	0.846

5.2.3 Generalization

Figure 5.16 provides additional examples showcasing our femur segmentation and 3D reconstruction results from unseen patients. These examples serve to demonstrate the performance and effectiveness of our proposed methodology on a diverse set of cases.

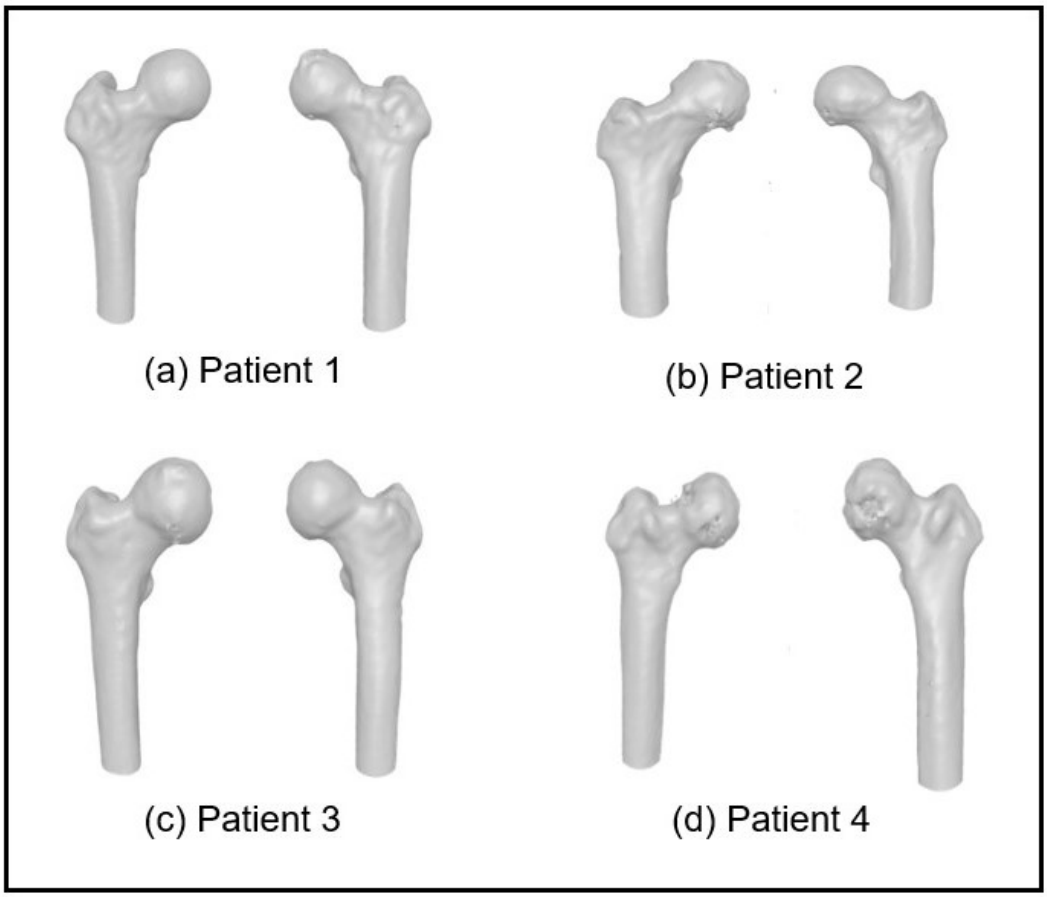


Figure 5.16: Examples of segmented and 3D reconstructed femurs of unseen patients by SSDL model.

These visual representations offer a comprehensive view of the accuracy and quality of our segmentation and reconstruction processes for femurs from previously unseen patients. By examining the figures, it is evident that our approach successfully captures the distinct anatomical

features of the femur, accurately delineating its boundaries and contours. Furthermore, the 3D femur reconstructions demonstrate the ability of our method to generate accurate and realistic representations of the femurs in three-dimensional space. The reconstructed models exhibit clear and well-defined surfaces, preserving the overall shape and morphology of the femurs. Overall, the examples presented in Figure 5.16 validate the robustness and generalizability of our approach for femur segmentation and 3D reconstruction. These results showcase our model's ability to effectively process and analyze QCT images from unseen patients, producing accurate and reliable outcomes across a range of different cases.

Our proposed model for segmentation to 3D reconstruction has the advantage of requiring significantly less time than the femur reconstruction by manual intervention with conventional image processing software (*3D Slicer*). The proposed framework requires approximately 85 ± 10 seconds to reconstruct the 3D bone once the QCT image dataset is loaded. On the other hand, the same process takes between 30-45 minutes by a well-trained undergraduate research assistant who has been working for nearly three years [23, 50]. Therefore, our model presents a more efficient and time-saving alternative to the existing conventional methods. The majority of previous studies in this field have not made available either the data, the models, or both, which poses a challenge for conducting a direct one-to-one comparison between our SSDL model and other existing models. It is also worth noting that, in the medical field, where the ultimate goal is clinical applications, it is uncommon to come across claims of state-of-the-art performance without a demonstrated history of clinical usage.

(A subject expert Dr. Ahmed Suparna Bahar Moni, Assistant Professor, Orthopedic Surgery, University of Toledo, USA took a look on reconstructed femur for validity.)

Chapter 6

Conclusion

In this chapter, we provide our research summary and future direction of extending this work.

6.1 Summary

In this research, we have presented an efficient, faster automated semi-supervised deep leaning-based model what we call SSDL for femur segmentation and 3D femur reconstruction from DICOM images directly. To minimize the cost and time required for data pre-processing, the proposed approach aimed at keeping the annotation effort to a minimum without compromising the annotation accuracy, which is a critical step for a supervised or semi-supervised framework. Only the QCT slices at the proximal end where multiple bones such as femur, acetabulum, and pelvis are in proximity have been annotated. The ability to preserve DICOM metadata interwoven with the segmented QCT images is one of the key aspects of this framework. By selecting a minimal number of slices to annotate and generating binary masks from these annotations, the proposed approach is able to leverage the power of deep learning while minimizing the effort required for data pre-processing. Overall, this approach offers a promising solution to the challenge of generating high-quality training datasets for medical image analysis, which is a crucial step in the development of accurate and effective automated analysis tools for clinical practice.

We have provided both of the qualitative and quantitative results to support the effectiveness of our proposed framework. Our framework has achieved a Dice Similarity Coefficient of 91.8% for unseen patients and 99.2% for validated patients. We have obtained an average Relative Error of 6.61% and 12.08% for volume and surface area, respectively. The proposed approach demonstrates its effectiveness in accurately segmenting and reconstructing the 3D femur from QCT slices.

6.2 Limitations and Challenges

At first, we would like to mention that, the 3D reconstruction of proximal femur remains a challenging task as it highly depends on the accurate annotation of QCT image datasets for training. The annotation process for labeling bone profiles on each QCT slice is time-consuming, expensive, and requires some level of expertise. Since the annotations are done manually, we encountered mismatch of the femur boundary due to conservative annotation, as well as difficulty in identifying the bone profiles, requiring redo of the annotation process. Although extreme care has been taken, a very small percentage of annotation errors may be anticipated in the training dataset, which might impact the performance of our framework. Secondly, finding an optimum segmentation threshold is difficult, and bone properties clustering does not yield good results [47]. Although our 3D U-Net framework has shown high segmentation performance with a very small number of patients, further improvement is still needed for a generalized femur segmentation system. We have noticed some negligible errors near the femoral head than the other parts of the femur. Due to conservative annotation, the model may predict femur head less clearly. Thus, a cascading error in the 3D reconstruction can be noted. We have further noticed that our framework in some cases shows suboptimal 3D femur reconstruction in proximal areas such as Fig. 5.16(d). We believe that this limitation will be solved with more accurately annotated femur slices for training dataset. One of major challenges in our 3D femur reconstruction framework is to determine optimum patient-specific threshold values for 3D femur reconstruction. These thresholding values also contribute to producing isolated islands during the 3D model generation. We need to conduct further analysis with more patients to observe the patterns before applying optimal threshold values. Thus, datasets with more patients and different races, ethnicity, medical conditions such as with and without osteoporosis should be included in both training and testing to increase its fidelity to work with a wide variety of patients.

6.3 Future Directions

In future, we want to continue improving our research for femur segmentation and 3D reconstruction. We are planning to extend the research as follows.

- We would like to optimize our proposed model SSDL to achieve faster and more accurately femur reconstruction as well as to evaluate the model's performance on a larger dataset of unseen patients.
- We aim to optimize and investigate the high error in Femur 4, depicted in Figure 5.15, by investigating the characteristics contributing to this significant deviation from the expected volume.

- We would like to explore the impact of various activation functions and other deep learning techniques on the femur segmentation of our SSDL model.
- We would like to investigate other machine learning and deep learning techniques for 3D femur reconstruction from the segmented femur slices.
- There is a severe lack of publicly available DL models and datasets for femur segmentation and 3D reconstruction. Hence, we also plan to explore unsupervised learning in future.

Our effort is expected to enhance the applicability of computational model in the clinical settings and to contribute to clinical research.

References

[1] A. Collo, "The hip joint: some hints," 2012. June 10, 2023.

[2] T. Gangwar, J. Calder, T. Takahashi, J. E. Bechtold, and D. Schillinger, "Robust variational segmentation of 3d bone ct data with thin cartilage interfaces," *Medical Image Analysis*, vol. 47, pp. 95–110, 2018.

[3] J. Stawiaski, E. Decenciere, and F. Bidault, "Interactive liver tumor segmentation using graph-cuts and watershed," in *11th international conference on medical image computing and computer assisted intervention (MICCAI 2008)*, 2008.

[4] M. Krčah, G. Székely, and R. Blanc, "Fully automatic and fast segmentation of the femur bone from 3d-ct images with no shape prior," in *2011 IEEE international symposium on biomedical imaging: from nano to macro*, pp. 2087–2090, IEEE, 2011.

[5] K. Koh, Y. H. Kim, K. Kim, and W. M. Park, "Reconstruction of patient-specific femurs using x-ray and sparse ct images," *Computers in biology and medicine*, vol. 41, no. 7, pp. 421–426, 2011.

[6] C. Chu, J. Bai, X. Wu, and G. Zheng, "Mascg: Multi-atlas segmentation constrained graph method for accurate segmentation of hip ct images," *Medical image analysis*, vol. 26, no. 1, pp. 173–184, 2015.

[7] Y. Xia, J. Fripp, S. S. Chandra, R. Schwarz, C. Engstrom, and S. Crozier, "Automated bone segmentation from large field of view 3d mr images of the hip joint," *Physics in Medicine & Biology*, vol. 58, no. 20, p. 7375, 2013.

[8] Y. Xia, S. S. Chandra, C. Engstrom, M. W. Strudwick, S. Crozier, and J. Fripp, "Automatic hip cartilage segmentation from 3d mr images using arc-weighted graph searching," *Physics in Medicine & Biology*, vol. 59, no. 23, p. 7245, 2014.

[9] V. C. Korfiatis, S. Tassani, and G. K. Matsopoulos, "An independent active contours segmentation framework for bone micro-ct images," *Computers in Biology and Medicine*, vol. 87, pp. 358–370, 2017.

[10] R. M. Y. Wong, P. Y. Wong, C. Liu, Y. L. Chung, K. C. Wong, C. Y. Tso, S. K.-H. Chow, W.-H. Cheung, P. S.-H. Yung, C. S. Chui, *et al.*, "3d printing in orthopaedic surgery: a scoping review of randomized controlled trials," *Bone & Joint Research*, vol. 10, no. 12, pp. 807–819, 2021.

[11] R. Ellis, C. Tso, J. Rudan, and M. Harrison, "A surgical planning and guidance system for high tibial osteotomy," *Computer Aided Surgery*, vol. 4, no. 5, pp. 264–274, 1999.

[12] I. Otomaru, M. Nakamoto, Y. Kagiyama, M. Takao, N. Sugano, N. Tomiyama, Y. Tada, and Y. Sato, "Automated preoperative planning of femoral stem in total hip arthroplasty from 3d ct data: Atlas-based approach and comparative study," *Medical Image Analysis*, vol. 16, no. 2, pp. 415–426, 2012.

[13] S. E. Sheehan, J. Y. Shyu, M. J. Weaver, A. D. Sodickson, and B. Khurana, "Proximal femoral fractures: what the orthopedic surgeon wants to know," *Radiographics*, vol. 35, no. 5, pp. 1563–1584, 2015.

[14] S. Siebenlist, T. Torsiglieri, T. Kraus, R. Burghardt, U. Stöckle, and M. Lucke, "Comminuted fractures of the proximal ulna—preliminary results with an anatomically preshaped locking compression plate (lcp) system," *Injury*, vol. 41, no. 12, pp. 1306–1311, 2010.

[15] K. Al-Hourani, S.-T. J. Tsang, and A. H. R. Simpson, "Osteoporosis: current screening methods, novel techniques, and preoperative assessment of bone mineral density," 2021.

[16] A. Gupta, C. Maslen, M. Vindlacheruvu, R. L. Abel, P. Bhattacharya, P. A. Bromiley, E. M. Clark, J. E. Compston, N. Crabtree, J. S. Gregory, *et al.*, "Digital health interventions for osteoporosis and post-fragility fracture care," *Therapeutic advances in musculoskeletal disease*, vol. 14, p. 1759720X221083523, 2022.

[17] G. M. Blake and I. Fogelman, "Role of dual-energy x-ray absorptiometry in the diagnosis and treatment of osteoporosis," *Journal of Clinical Densitometry*, vol. 10, no. 1, pp. 102–110, 2007.

[18] B. Grignon, G. Oldrini, and F. Walter, "Teaching medical anatomy: what is the role of imaging today?," *Surgical and Radiologic Anatomy*, vol. 38, pp. 253–260, 2016.

[19] H. Kasban, M. El-Bendary, and D. Salama, "A comparative study of medical imaging techniques," *International Journal of Information Science and Intelligent System*, vol. 4, no. 2, pp. 37–58, 2015.

[20] K. Ohnaru, T. Sone, K. Tanaka, K. Akagi, Y.-I. Ju, H.-J. Choi, T. Tomomitsu, and M. Fukunaga, "Hip structural analysis: a comparison of dxa with ct in postmenopausal japanese women," *Springerplus*, vol. 2, no. 1, pp. 1–8, 2013.

[21] L. Anez-Bustillos, L. C. Derikx, N. Verdonschot, N. Calderon, D. Zurakowski, B. D. Snyder, A. Nazarian, and E. Tanck, "Finite element analysis and ct-based structural rigidity analysis to assess failure load in bones with simulated lytic defects," *Bone*, vol. 58, pp. 160–167, 2014.

[22] E. Benca, A. Synek, M. Amini, F. Kainberger, L. Hirtler, R. Windhager, W. Mayr, and D. H. Pahr, "Qct-based finite element prediction of pathologic fractures in proximal femora with metastatic lesions," *Scientific reports*, vol. 9, no. 1, p. 10305, 2019.

[23] T. R. Faisal and Y. Luo, "Study of the variations of fall induced hip fracture risk between right and left femurs using ct-based fea," *Biomedical engineering online*, vol. 16, pp. 1–17, 2017.

[24] Y. Lee, N. Ogihara, and T. Lee, "Assessment of finite element models for prediction of osteoporotic fracture," *Journal of the mechanical behavior of biomedical materials*, vol. 97, pp. 312–320, 2019.

[25] M. Mirzaei, M. Keshavarzian, and V. Naeini, "Analysis of strength and failure pattern of human proximal femur using quantitative computed tomography (qct)-based finite element method," *Bone*, vol. 64, pp. 108–114, 2014.

[26] L. Ben Younes, Y. Nakajima, and T. Saito, "Fully automatic segmentation of the femur from 3d-ct images using primitive shape recognition and statistical shape models," *International journal of computer assisted radiology and surgery*, vol. 9, pp. 189–196, 2014.

[27] J. Carballido-Gamio, S. Bonaretti, I. Saeed, R. Harnish, R. Recker, A. J. Burghardt, J. H. Keyak, T. Harris, S. Khosla, and T. F. Lang, "Automatic multi-parametric quantification of the proximal femur with quantitative computed tomography," *Quantitative imaging in medicine and surgery*, vol. 5, no. 4, p. 552, 2015.

[28] S. S. Chandra, Y. Xia, C. Engstrom, S. Crozier, R. Schwarz, and J. Fripp, "Focused shape models for hip joint segmentation in 3d magnetic resonance images," *Medical image analysis*, vol. 18, no. 3, pp. 567–578, 2014.

[29] B. Gilles and N. Magnenat-Thalmann, "Musculoskeletal mri segmentation using multi-resolution simplex meshes with medial representations," *Medical image analysis*, vol. 14, no. 3, pp. 291–302, 2010.

[30] M. Kardell, M. Magnusson, M. Sandborg, G. Alm Carlsson, J. Jeuthe, and A. Malusek, "Automatic segmentation of pelvis for brachytherapy of prostate," *Radiation Protection Dosimetry*, vol. 169, no. 1-4, pp. 398–404, 2016.

[31] S. Vasilache and K. Najarian, "Automated bone segmentation from pelvic ct images," in *2008 IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pp. 41–47, IEEE, 2008.

[32] P. Kalshetti, M. Bundele, P. Rahangdale, D. Jangra, C. Chattopadhyay, G. Harit, and A. Elhence, "An interactive medical image segmentation framework using iterative refinement," *Computers in biology and medicine*, vol. 83, pp. 22–33, 2017.

[33] A. Malusek, M. Magnusson, M. Sandborg, and G. Alm Carlsson, "A model-based iterative reconstruction algorithm dira using patient-specific tissue classification via dect for improved quantitative ct in dose planning," *Medical Physics*, vol. 44, no. 6, pp. 2345–2357, 2017.

[34] F. Lopez-Jimenez, Z. Attia, A. M. Arruda-Olson, R. Carter, P. Chareonthaitawee, H. Jouni, S. Kapa, A. Lerman, C. Luong, J. R. Medina-Inojosa, *et al.*, "Artificial intelligence in cardiology: present and future," *Mayo Clinic Proceedings*, vol. 95, no. 5, pp. 1015–1039, 2020.

[35] C. A. Peña-Solórzano, D. W. Albrecht, R. Bassed, J. Gillam, P. Harris, and M. Dimmock, "Semi-supervised labelling of the femur in a whole-body post-mortem ct database using deep learning," *Computers in Biology and Medicine*, vol. 122, p. 103797, 2020.

[36] D. Rokaya, S. Kongkiatkamon, A. Heboyan, V. V. Dam, P. Amornvit, Z. Khurshid, V. Srimaneepong, and M. S. Zafar, "3d-printed biomaterials in biomedical application," in *Functional Biomaterials: Drug Delivery and Biomedical Applications*, pp. 319–339, Springer, 2022.

[37] P. A. Bjornsson, B. Helgason, H. Palsson, S. Sigurdsson, V. Gudnason, and L. M. Ellingsen, "Automated femur segmentation from computed tomography images using a deep neural network," in *Medical Imaging 2021: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 11600, pp. 324–330, SPIE, 2021.

[38] C. Wang, N. Dahlström, S.-G. Fransson, C. Lundström, *et al.*, "Real-time interactive 3d tumor segmentation using a fast level-set algorithm," *Journal of Medical Imaging and Health Informatics*, vol. 5, no. 8, pp. 1998–2002, 2015.

[39] C. Zhao, J. H. Keyak, J. Tang, T. S. Kaneko, S. Khosla, S. Amin, E. J. Atkinson, L.-J. Zhao, M. J. Serou, C. Zhang, *et al.*, "A deep learning-based method for automatic segmentation of proximal femur from quantitative computed tomography images," *arXiv preprint arXiv:2006.05513*, 2020.

[40] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and*

Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19, pp. 424–432, Springer, 2016.

[41] J. C. G. Sánchez, M. Magnusson, M. Sandborg, Å. C. Tedgren, and A. Malusek, "Segmentation of bones in medical dual-energy computed tomography volumes using the 3d u-net," *Physica Medica*, vol. 69, pp. 241–247, 2020.

[42] A. Klein, J. Warszawski, J. Hillengaß, and K. H. Maier-Hein, "Automatic bone segmentation in whole-body ct images," *International journal of computer assisted radiology and surgery*, vol. 14, pp. 21–29, 2019.

[43] S. Noguchi, M. Nishio, M. Yakami, K. Nakagomi, and K. Togashi, "Bone segmentation on whole-body ct using convolutional neural network with novel data augmentation techniques," *Computers in biology and medicine*, vol. 121, p. 103767, 2020.

[44] F. Chen, J. Liu, Z. Zhao, M. Zhu, and H. Liao, "Three-dimensional feature-enhanced network for automatic femur segmentation," *IEEE journal of biomedical and health informatics*, vol. 23, no. 1, pp. 243–252, 2017.

[45] Y. Deng, L. Wang, C. Zhao, S. Tang, X. Cheng, H.-W. Deng, and W. Zhou, "A deep learning-based approach to automatic proximal femur segmentation in quantitative ct images," *Medical & biological engineering & computing*, vol. 60, no. 5, pp. 1417–1429, 2022.

[46] L. Zhu, J. Han, R. Guo, D. Wu, Q. Wei, W. Chai, and S. Tang, "An automatic classification of the early osteonecrosis of femoral head with deep learning," *Current medical imaging*, vol. 16, no. 10, pp. 1323–1331, 2020.

[47] J. Jeuthe, J. C. G. Sánchez, M. Magnusson, M. Sandborg, Å. C. Tedgren, and A. Malusek, "Semi-automated 3d segmentation of pelvic region bones in ct volumes for the annotation of machine learning datasets," *Radiation Protection Dosimetry*, vol. 195, no. 3-4, pp. 172–176, 2021.

[48] J. Stawiaski, E. Decenciere, and F. Bidault, "Computing approximate geodesics and minimal surfaces using watershed and graph cuts.," in *ISMM (1)*, pp. 349–360, 2007.

[49] T. R. Faisal and Y. Luo, "Study of stress variations in single-stance and sideways fall using image-based finite element analysis," *Bio-medical materials and engineering*, vol. 27, no. 1, pp. 1–14, 2016.

[50] R. Awal, J. Ben Hmida, Y. Luo, and T. Faisal, "Study of the significance of parameters and their interaction on assessing femoral fracture risk by quantitative statistical analysis," *Medical & biological engineering & computing*, vol. 60, no. 3, pp. 843–854, 2022.

[51] S. Khan, R. Warkhedkar, and A. Shyam, "Analysis of hounsfield unit of human bones for strength evaluation," *Procedia Materials Science*, vol. 6, pp. 512–519, 2014.

[52] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[57] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, 2022.

[58] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019.

[59] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge 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: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pp. 240–248, Springer, 2017.

[60] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[61] K. M. Ting, *Sensitivity and Specificity*. Boston, MA: Springer US, 2010.

[62] C. Sammut and G. I. Webb, "Leave-one-out cross-validation," *Encyclopedia of machine learning*, pp. 600–601, 2010.

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

[64] G. Van Rossum and F. L. Drake, *Introduction to python 3: python documentation manual part 1*. CreateSpace, 2009.

[65] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[66] D. Mason, "Su-e-t-33: pydicom: an open source dicom library," *Medical Physics*, vol. 38, no. 6(10), pp. 3493–3493, 2011.

[67] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[68] G. Wolberg and I. Alfy, "An energy-minimization framework for monotonic cubic spline interpolation," *Journal of Computational and Applied Mathematics*, vol. 143, no. 2, pp. 145–188, 2002.

[69] M. Musy, G. Jacquenot, G. Dalmasso, R. de Bruin, A. Pollack, F. Claudi, C. Badger, B. Sullivan, D. Hrisca, D. Volpatto, *et al.*, "vedo: A python module for scientific analysis and visualization of 3d objects and point clouds," *Zenodo*, 2021.

[70] A. Chaudhary, S. J. Jhaveri, A. Sanchez, L. S. Avila, K. M. Martin, A. Vacanti, M. D. Hanwell, and W. Schroeder, "Cross-platform ubiquitous volume rendering using programmable shaders in vtk for scientific and medical visualization," *IEEE Computer Graphics and Applications*, vol. 39, no. 1, pp. 26–43, 2019.

[71] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, p. 163–169, aug 1987.

[72] Y. Tan, J. Hua, and M. Dong, "3d reconstruction from 2d images with hierarchical continuous simplices," *The Visual Computer*, vol. 23, pp. 905–914, 2007.

[73] F. Farheen, M. S. Shamil, N. Ibtehaz, and M. S. Rahman, "Revisiting segmentation of lung tumors from ct images," *Computers in Biology and Medicine*, vol. 144, p. 105385, 2022.

Appendix A

Segmented QCT Images

A.1 Sample of proximal femur segmentation from QCT images by SSDL

We have attached some samples of input-output QCT scan pair for visualization. These samples are reviewed by a domain expert for their correctness.

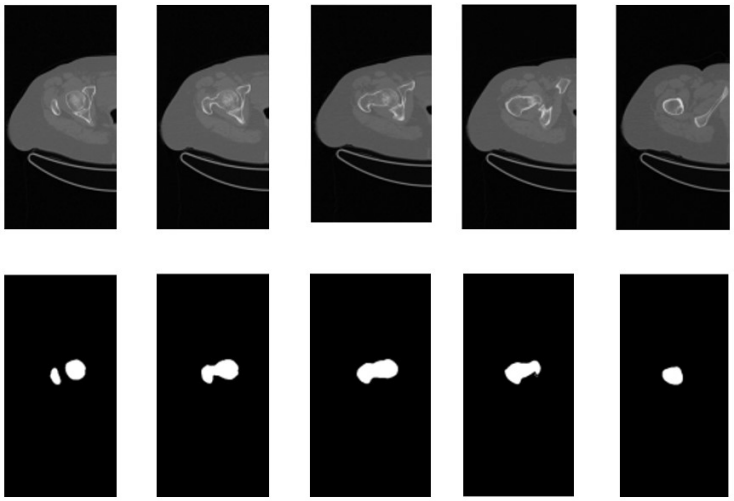


Figure A.1: Right proximal femur segmentation from QCT images of Patient 1.

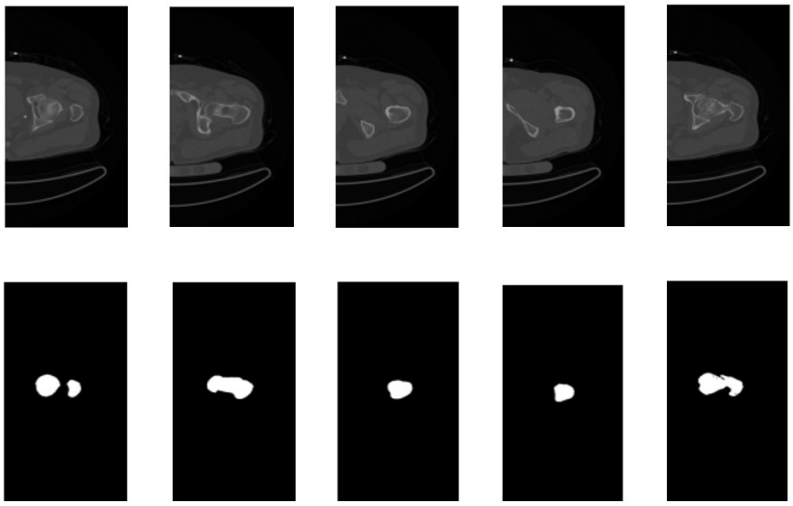


Figure A.2: Left proximal femur segmentation from QCT images of Patient 1.

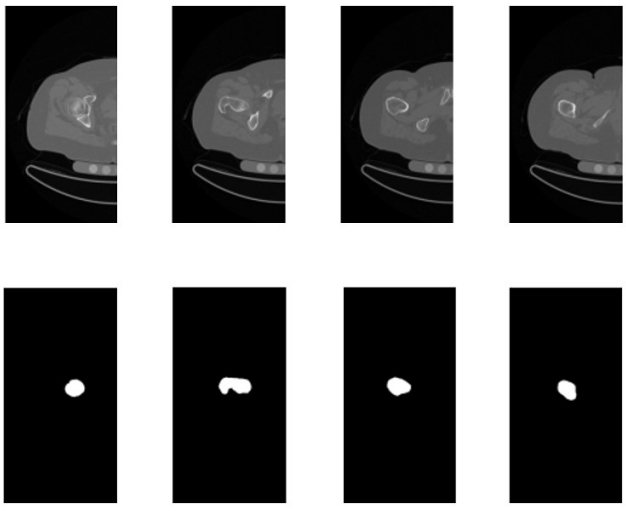


Figure A.3: Right proximal femur segmentation from QCT images of Patient 2.

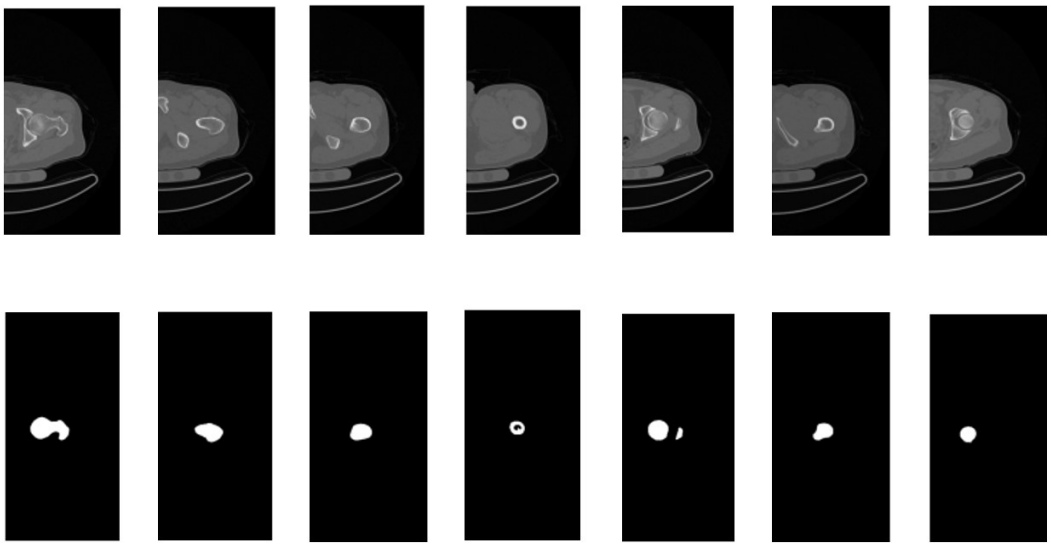


Figure A.4: Left proximal femur segmentation from QCT images of Patient 2.

Generated using Postgraduate Thesis LaTeX Template, Version 1.03. Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

This thesis was generated on Saturday 24th June, 2023 at 3:50am.