FACULTY OF ENGINEERING AND ARCHITECTURE

Lower Limb Bone Segmentation through Deep Learning and Neural Flow Based Data Augmentation

Roel Huysentruyt
Student number: 01710473

Supervisors: Prof. dr. ir. Aleksandra Pizurica, Prof. dr. Emmanuel Audenaert
Counsellors: Srdan Lazendic, Ide Van den Borre

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering

Academic year 2022-2023

GHENT UNIVERSITY

VUB VRIJE UNIVERSITEIT BRUSSEL

FACULTY OF ENGINEERING AND ARCHITECTURE

Lower Limb Bone Segmentation through Deep Learning and Neural Flow Based Data Augmentation

Roel Huysentruyt
Student number: 01710473

Supervisors: Prof. dr. ir. Aleksandra Pizurica, Prof. dr. Emmanuel Audenaert
Counsellors: Srdan Lazendic, Ide Van den Borre

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering

Academic year 2022-2023

GHENT UNIVERSITY

VUB VRIJE UNIVERSITEIT BRUSSEL

Permission of use on loan

The author gives permission to make this master's dissertation available for consultation and to copy parts of this master's dissertation for personal use. In all cases of other use, the copyright terms have to be respected, in particular with regard to the obligation to state explicitly the source when quoting results from this master dissertation.

ROEL HUYSENTRUYT
June 5, 2023

Preface

This master's dissertation represents the end of my studies in Biomedical Engineering and the beginning of more research in this interesting field. Throughout the years, my passion for medical imaging and the application of machine learning has grown significantly. This research endeavour provided me with the opportunity to delve deeper into these interests, applying machine learning to a real-life scenario. The ultimate goal was to contribute to the development of a practical tool that can facilitate future research efforts. Just like in any endeavour, achieving something is never solely attributed to oneself. Therefore, I would like to express my gratitude towards certain individuals.

First and foremost, I would like to express my gratitude to my promotors, prof. dr. ir. Aleksandra Pižurica and prof. dr. Emmanuel Audenaert, for both providing me with this interesting research topic and consistently demonstrating their interest in my progress throughout the year.

I also want to thank Srdan Lazendić and ir. Ide Van den Borre for their guidance during this project, Srdan provided valuable insights and ideas, and Ide demonstrated exceptional cooperation, always readily available to address any questions or concerns that arose.

Next, I would like to thank my family for the opportunities they provided me with and the unconditional support throughout my entire educational journey. Last but not least, I want to thank my friends. I have enjoyed every second of these past years thanks to you. First, I would like to express appreciation to the friends I have made in this field of study. Their openness to discussions and their ability to provide much-needed laughter when it was necessary, have been truly invaluable. Furthermore, I would like to extend my gratitude to the friends who have always been there to share in the enjoyment of my free time, getting drinks and creating unforgettable memories. It is with you that I have experienced the most joyous moments of my life thus far, and I am sincerely grateful for your presence, as every single moment of this journey has been enriched by your friendship.

ROEL HUYSENTRUYT

Lower Limb Bone Segmentation through Deep Learning and Neural Flow Based Data Augmentation

Roel Huysentruyt

Supervisor: Prof. dr. ir. Aleksandra Pižurica + Prof. dr. Emmanuel Audenaert
Counsellors: Srđan Lazendić + ir. Ide Van den Borre

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering
Academic year 2022-2023

Abstract

With the increasing availability of labelled data, advancements in computing power, and continuous technical improvements, the use of convolutional neural networks (CNNs) in the development of automatic segmentation methods has witnessed a remarkable increase. However, a lot of research showing promising results utilising CNNs focus on a specific region of interest (ROI). Furthermore, existing approaches for whole-body computed tomography (CT) images, including both traditional methods and deep learning (DL) based techniques, suffer from either compromised accuracy or inadequate speed. This study presents a novel workflow, which is simple yet fast and accurate, to segment the major bones of the lower limb region on a whole-body CT scan, thereby broadening the scope of analysis beyond ROIs and improving both accuracy and speed. The use of these bone segmentations can enhance the diagnosis and assessment of bone diseases, aid in treatment planning, and promote further research. This thesis specifically concentrates on creating bone segmentations from healthy patients, aiming to facilitate further research efforts. The developed workflow encompasses two stages, both leveraging the 3D-UNet architecture. In the initial step, the workflow facilitates the localisation of the ROIs, while the subsequent stage focuses on achieving precise segmentations. With a processing time of under 15 seconds, this method efficiently analyses whole-body CT scans, yielding precise segmentations of the pelvis, femur, tibia, and fibula. The method achieves an average Dice similarity coefficient (DSC) of 0.978 outperforming most of the state-of-the-art methods. Results are further evaluated and discussed from a technical and clinical viewpoint. In addition to the segmentation workflow, a novel data augmentation technique has been developed to generate new shape and image pairs that adhere to population statistics. This method leverages FlowSSM to generate nonaligned shapes and incorporates a TSP warp to create corresponding medical images. Although this augmentation technique was not specifically applied in our segmentation application, future DL methods could potentially benefit from its implementation.

Index Terms: Segmentation, deep learning, computed tomography, data augmentation

Lower Limb Bone Segmentation through Deep Learning and Neural Flow Based Data Augmentation

Roel Huysentruyt
Supervisor: Prof. dr. ir. Aleksandra Pižurica and Prof. dr. Emmanuel Audenaert
Counsellors: Srđan Lazendić and ir. Ide Van den Borre

***Abstract*—With the increasing availability of labelled data, advancements in computing power, and continuous technical improvements, the use of convolutional neural networks (CNNs) in the development of automatic segmentation methods has witnessed a remarkable increase. However, a lot of research showing promising results utilising CNNs focus on a specific region of interest (ROI). Furthermore, existing approaches for whole-body computed tomography (CT) images, including both traditional methods and deep learling (DL) based techniques, suffer from either compromised accuracy or inadequate speed. This study presents a novel workflow, which is simple yet fast and accurate, to segment the major bones of the lower limb region on a whole-body CT scan, thereby broadening the scope of analysis beyond ROIs and improving both accuracy and speed. The use of these bone segmentations can enhance the diagnosis and assessment of bone diseases, aid in treatment planning, and promote further research. This thesis specifically concentrates on creating bone segmentations from healthy patients, aiming to facilitate further research efforts. The developed workflow encompasses two stages, both leveraging the 3D-UNet architecture. In the initial step, the workflow facilitates the localisation of the ROIs, while the subsequent stage focuses on achieving precise segmentations. With a processing time of under 15 seconds, this method efficiently analyses whole-body CT scans, yielding precise segmentations of the pelvis, femur, tibia, and fibula. The method achieves an average Dice similarity coefficient (DSC) of 0.978 outperforming most of the state-of-the-art methods. Results are further evaluated and discussed from a technical and clinical viewpoint. In addition to the segmentation workflow, a novel data augmentation technique has been developed to generate new shape and image pairs that adhere to population statistics. This method leverages FlowSSM to generate nonaligned shapes and incorporates a TSP warp to create corresponding medical images. Although this augmentation technique was not specifically applied in our segmentation application, future DL methods could potentially benefit from its implementation.**

***Index Terms*—Segmentation, deep learning, computed tomography, data augmentation**

I. Introduction

Advancements in orthopaedic and biomechanical research heavily rely on image segmentation of the bones in the lower limb region. However, the manual delineation of anatomical structures on medical images for segmentation is a time-consuming and labor-intensive process [1]. For instance, even delineating the femur and pelvis alone can take up to 10 hours for a single case [2]. To overcome these limitations, there is a growing need for fully automated methods. While atlas-based methods and using statistical shape models (SSM) that rely on prior knowledge have shown promising results [1], [3], [4], they still leave room for improvement in terms of segmentation accuracy and inference speed on new cases. For instance, the automatic segmentation of a full lower limb case with a statistical shape model method can still take up to 2 hours.

The use of deep learning (DL) for medical applications has seen a significant surge due to the increase in computing power, widespread availability of software, continuous improvements, and the abundance of data [5], [6]. Compared to more traditional techniques, DL techniques can offer fast and accurate predictions, making them advantageous. Particularly, Convolutional Neural Networks (CNNs) have emerged as a powerful tool for interpreting medical imaging data, also for segmentation. The UNet architecture, introduced in 2015 by Ronnberger et al., has further propelled this trend towards using DL for medical image segmentation [7].

While recent research has yielded promising results in the area of bone segmentation, many studies focus on datasets that are centred around the region of interest (ROI) or have already been cropped to it [8]–[10]. Methods that operate on whole-body computed tomography (CT) scans often suffer from lower accuracy or slow inference times when applied to unseen cases [11], [12]. Moreover, these methods face additional challenges due to the variations in available data. Each network is trained and deployed on a distinct dataset, which can lead to issues when applying the model to different datasets. This makes it difficult to compare techniques directly. Hence, a standard method that achieves both accurate and fast predictions is still not readily available. The objective of this thesis is to develop a robust and accurate subject-specific segmentation workflow for the lower limb bones, which also provides fast predictions. Specifically, the workflow should have the ability to process a whole-body CT scan without requiring manual cropping to specific ROIs. This approach will be created and evaluated on an in-house dataset of whole-body CT scans of healthy patients, with the expectation that it will outperform the current state-of-the-art method in terms of both speed and accuracy. By integrating these ideas with upcoming research, a valuable tool can be developed that serves as the

initial step for future orthopaedic studies that require subject-specific segmentations. One of the main challenges of many DL approaches is the lack of sufficient data for training. Therefore, alongside the segmentation method, a novel data augmentation technique is developed to address this issue. Although this augmentation method is not currently used in combination with the segmentation workflow, it may prove beneficial in future applications. The primary goal of this augmentation approach is to generate new shape and image pairs that conform to the modelled population. This method is based on the data augmentation created by the authors of DeepSSM, developed by ShapeWorks [13], [14]. However, with the help of FlowSSM, a new way of generating shapes is implemented that does not require any alignment and can model shape nonlinearities [15].

II. Dataset

DL requires a significant amount of annotated data to achieve optimal performance and generalise well to new cases. Therefore, having a robust dataset is essential to train, deploy, and test deep learning models accurately, particularly when dealing with accurate segmentation. In this study, a dataset that includes information from 94 healthy subjects, each of whom had a full-body CT image and 3D bone models available. Out of the 94 subjects, 53 are male, 40 are female, and 1 is nonbinary, and their ages ranged from 39 to 96 years, with an average age of 66.7 years and a standard deviation of 14.4 years. Most of the images were acquired at AZ Groeninge in Kortrijk, Belgium using a GE medical system, although the dataset is not restricted to this hospital. The slice thickness of the images, which reflects the resolution in the z-direction, is either 0.625mm or 1.250mm. Spacing in the x- and y-direction varies in the range between 0.578mm and 0.971mm. Models of the pelvis, femur, tibia, and fibula are available as meshes, with separate files provided for the left and right sides, resulting in a total of 6 files. The pelvis is originally only available as one model but is preprocessed to split the left and right hip bones. The average edge length of the models vary between 1.14mm and 1.77mm.

III. Proposed methods

A. Segmentation workflow

1) General workflow overview: The workflow utilises the UNet model for segmentation, but due to computational limitations, the full CT scans cannot be processed at once. Instead, the workflow suggests working with sub-images. The process involves locating the general positions of the bones in the downsampled CT image, cropping the full-resolution image to the ROI, and training a UNet on these downsampled images for multiclass segmentation. The predictions on the downsampled image allow finding the bounding boxes of the bones in the full-resolution image, enabling the extraction of sub-images focused on each bone. Separate UNet networks are trained and applied to these sub-images for final predictions, which can be merged back into the original CT image. The workflow prioritises generating accurate segmentation maps, which can later be converted into 3D models.

2) Data preprocessing: Training the UNet requires label maps, which serve as ground truth data for accurate segmentation. Since the available data includes 3D bone models, a conversion method is needed to create label maps from these models. Each subject has a label map with values ranging from 0 to 6, representing different structures. A Matlab script, using the mesh voxelisation method from Adam. A., is used to create the label maps by taking CT images and 3D models as input [16]. Before creating the label maps, the voxel size of the CT images is standardised to 1mm using linear interpolation. This helps standardise the data for more stable segmentations with a CNN. For the first step, identifying the ROIs, the images are further downsized by sampling every 5th voxel in the z-direction and every 4th voxel in the x- and y-directions. For the further segmentations of the ROI, cropped versions of the image without downsampling are used. Image padding is applied to all images, both the downsampled full CT as the sub-images focusing on the ROI, to make the image size divisible by 8 in each direction to accommodate the UNet's down- and upsampling operations. Padding involves adding voxels with a value of -1024.0 around the image. Normalisation is performed by subtracting the mean of the image and dividing by the standard deviation, bringing the data to a standardised range.

3) UNet architecture: The UNet architecture utilised in this study consists of an encoder, a decoder, and a bottleneck block. The encoder comprises three convolutional blocks with instance normalisation and ReLU activation, while the decoder incorporates three upsampling blocks with transposed convolutional layers. The feature maps from the encoder are concatenated with the decoder blocks, following the typical UNet approach. The concatenated feature map undergoes a convolutional block similar to the encoder, and the output of the decoder is passed through a final 3D convolutional layer for segmentation. This network is inspired by the work of Ambellan et al. (2019), but with modifications such as the absence of dropout and fewer convolutional filters, resulting in comparable performance while reducing memory usage. The network is shown in Fig. 1.

4) Training procedure: The training procedure is consistent across all networks. One network is trained on downsampled CT images, while three networks are trained on sub-images of the pelvis, femur, tibia, and fibula combined. The dataset is randomly divided into three groups: a training group with 72 subjects, a validation group with 10 subjects, and a testing group with 9 subjects, which is used for evaluating the model's performance on new cases. The UNet parameters are initialised randomly and updated using the Adam optimiser. The network trained on downsampled images utilises a weighted cross-entropy loss (WCE) to handle the class-imbalance, with a weight of 1/10 assigned to the background

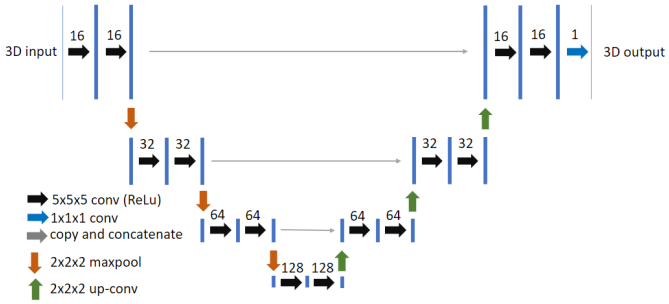


Fig. 1: UNet architecture.

class and a weight of 1 for bones. The networks trained on sub-images employ binary cross-entropy loss (BCE). Hyperparameters are manually fine-tuned, with a learning rate of 0.001 and a batch size of 1 to mitigate memory issues when dealing with large images. The amount of epochs depends on the case itself.

5) Model evaluation: The final segmentation maps are evaluated both visually and quantitatively using four metrics. Each metric is calculated separately for each bone. The classification of each voxel falls into one of the following categories: true positive (TP), true negative (TN), false positive (FP), false negative (FN), and predicted voxels. Three of the metrics, namely precision, recall, and Dice similarity coefficient (DSC), are defined based on these categories by

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3}$$

The final metric used is the Hausdorff distance (HD), an indicator of the biggest segmentation error. It can be seen as the furthest distance between a point in one of the two sets to its closest point in the other one and is defined as

$$\text{HD}(A, B) = \max(hd(A, B), hd(B, A))), \tag{4}$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||, \tag{5}$$

An extra evaluation is added by converted the segmentation maps to 3D models using a marching cubes algorithm, allowing further visual evaluation. An extra Taubin smoothing with parameters is applied $\lambda = 0.8$ and $\mu = -0.54$.

B. Neural-flow data augmentation

1) Method: The proposed data augmentation method is inspired by the one proposed by the authors of DeepSSM [14]. The goal is to generate new shape and image pairs. A novel method is introduced to overcome the limitations of alignment requirements and the inability to generate nonlinear shape variations in DeepSSM, which relies on PCA. The key distinction between the proposed method and DeepSSM's data augmentation lies in the shape generation approach. Instead of sampling the PCA space, a different method capable is employed: FlowSSM. The subsequent step involves determining the warp of each generated mesh to its nearest training mesh, followed by applying the same warps to the corresponding training image. The novelty of this method lies in utilising FlowSSM for shape generation. The proposed method offers a fresh approach to generating shape-image samples that effectively deal with shape nonlinearities and eliminate the need for alignment. Consequently, it results in a more diverse and representative dataset, enhancing the performance of deep learning techniques. This thesis encompasses two completed use cases. The first case focuses on testing a simple shape, specifically the distal femur. Subsequently, a more challenging shape, namely the pelvis, is employed in the second case.

2) Data preprocessing: Some preprocessing steps are applied to the meshes and images before doing the augmentation. The used images are the interpolated CTs obtained in the segmentation preprocessing with a spacing of 1mm. In the case of the distal femur, the meshes are cropped manually, reducing the number of vertices and faces. For both the femur and pelvis, centring of the meshes is performed by translating them to have a mean coordinate of (0,0,0). The same translation is applied to the CT images to maintain correspondence. Further alignment is not applied, as the aim is to overcome the alignment requirement limitation of DeepSSM. The images are then cropped to a region of interest based on a general bounding box created from all meshes, resulting in the same image size for each sample. After these preprocessing steps, the shape and image pairs are ready for the novel training augmentation method.

3) FlowSSM shape generation: To generate new shapes that adhere to the population statistics without the need for alignment, FlowSSM is used [15]. FlowSSM is an advanced technique for creating a neural flow-based Statistical Shape Model (SSM) that accurately represents natural shape variation. Unlike traditional SSMs that rely on PCA and require correspondences, FlowSSM uses multi-layer perceptrons (MLPs) to capture shape deformations without the need for correspondences. In FlowSSM, a specific type of shape is modelled by training a decoder that can produce a continuous flow to deform a template shape into a target shape. The decoder learns continuous deformations of the template surface to the target surface. During training, both the flow and latent representations are optimised, resulting in a latent space that accurately represents the population. This approach generates latent shape representations for each shape, allowing for population description in a latent space. The used template is the mean shape for the pelvis and the shape closest to the mean for the distal femur.

FlowSSM uses two types of shape representations: global and local. While a global representation produces smooth and low-frequency deformations, including a local parametrisation allows for high-frequency deformations. To achieve this, two networks or deformers are used: a global deformer for

Parameter	Distal femur	Pelvis
train epochs	100	300
train lr	0.001	0.001
embedding epochs	250	350
embedding lr	0.01	0.01
batch size	8	4
size latent embeddings	64	128
RBF size	7	6

TABLE I: Labels and their corresponding structure.

low-frequency deformation and a local deformer for high-frequency deformation.

FlowSSM requires a number of hyperparameters. For the use cases of the distal femur and the pelvis, these are given in Table I. For more information about these, please refer to the FlowSSM documentation. The loss function optimised during training of the deformers is the Chamfer distance (CD) given by

$$CD(P_i, P_\Phi) = \frac{0.5}{2|P_i|} \sum_{x_i \in P_i} \min_{x \in P_\Phi} \|x_i - x\|_2 + \frac{0.5}{2|P_\Phi|} \sum_{x \in P_\Phi} \min_{x_i \in P_i} \|x_i - x\|_2 \quad (6)$$

calculated between sampled surface points of the target $P_i \subset X_i$ and the corresponding deformed surface point of the template P_Φ. The deformation flow is found by minimising this over the complete training data. To generate new shapes, sampling is performed separately for the global and local embeddings. These sampled embeddings are used to deform the template shape. In this work, 1000 shapes are sampled this way.

4) TSP warp: Once shapes are generated, they are used to generate the corresponding CT images. This is done in the same way as in DeepSSM using the TSP warp developed by ShapeWorks. Note that using the TSP warp requires corresponding shapes. Even though FlowSSM does not require correspondence of the shapes, completing the data augmentation with the TSP warp does. The TSP warp is found from the closest original training shape to the generated shape. The closest training shape is found by the smallest average point-to-point Euclidean distance.

5) Evaluation: The results of FlowSSM are evaluated in two ways: generalisability and sensitivity. First the ability of the trained deformers to generalise on unseen cases is checked by calculating the CD and HD on the test set. A small distance shows the ability to generalise, also indicating that the latent space is constructed correctly.

Next, the specificity of the model is checked by calculating the average CD between each generated shape and its closest training shape. The specificity of a model is obtained as the average CD over 1000 generated shapes. The shapes are aligned to a template for a fair comparison, focusing on the shape itself rather than the orientation. This means that for the specificity measurement the shapes are aligned and we're able to compare to the results obtained using the PCA method from DeepSSM.

IV. Experimental results and discussion

A. Segmentation workflow

1) Training and predictions: The models were trained on a server equipped with a Tesla V100-PCIE-16GB GPU at the University Hospital of Ghent (UZ Gent). Each epoch had an average duration of 1200 seconds. The training of the femur model required the longest training time, consisting of 60 epochs, but achieved the lowest loss value of 0.00477. The other models had increasing final validation loss values, with 0.00724 for the pelvis, 0.00760 for the downsampled training, and 0.0107 for the tibia combined with the fibula model. The femur combined with the tibia model was trained for 40 epochs, while the pelvis and downsampled models were trained for 50 epochs. A prediction on a new unseen case from full CT to a segmentation map takes around 15s.

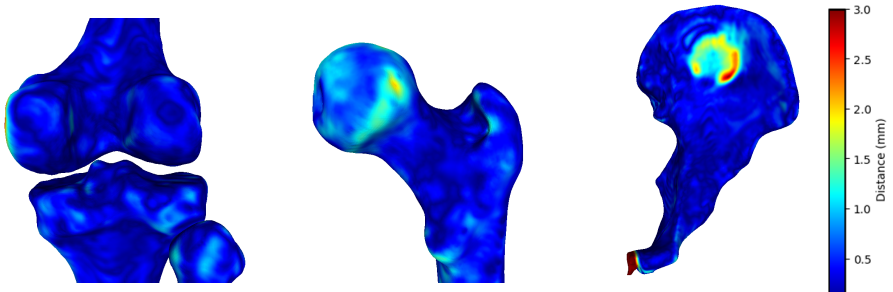
2) Evaluation metrics: The evaluation metrics for bone segmentation are presented in Table II. The metrics, including DSC, recall, and precision, exhibit high mean values with low standard deviations for all bone classes, indicating good performance across all unseen cases. The appendix provides a comprehensive breakdown of the metrics for each bone class per subject. Notably, the pelvis shows the lowest DSC values with means of 0.972 and 0.974, while the femur performs the best with a DSC of 0.985. Similar trends are observed for precision and recall.

Despite the high DSC, precision, and recall, which suggest accurate predictions, the HD appears to be high in many cases. This may be attributed to the model's inability to capture intricate details or variations in object shapes. However, because the other metrics are very high it could also be due to a few misclassified voxels at larger distance from the shapes. The left pelvis exhibits the highest HD values.

While label masks may resemble the ground truth, certain errors are more effectively identified by analysing 3D models. To visualise these errors, we calculate the minimal distance from each vertex of the prediction to the ground truth model. This information is then utilised in a colour-coded system to indicate the severity of the error. Fig.2 showcase the results for one subject N14, highlighting the challenges faced in the prediction process. It's important to note that these examples includes the worst results for the pelvis. The predictions for the proximal femur and knee joint exhibit low error rates, with only a few spots showing errors, typically within a range of 2mm. However, in the case of the pelvis, we observe a leakage issue at a specific location. This leakage is in some cases also seen at the distal femur. Overall, the majority of predictions demonstrate a close alignment with the ground truth, as evidenced by the high values of Dice Similarity Coefficient (DSC), precision, and recall, which indicate excellent segmentations. However, it is important to acknowledge that a few errors persist, such as the observed leakage. Additionally,

Type	DSC	Precision	Recall	HD (mm)
Femur R	0.985 ± 0.002	0.989 ± 0.003	0.981 ± 0.005	3.86 ± 2.94
Femur L	0.985 ± 0.004	0.989 ± 0.003	0.981 ± 0.007	4.78 ± 2.60
Pelvis R	0.974 ± 0.002	0.979 ± 0.005	0.969 ± 0.004	7.97± 5.16
Pelvis L	0.972 ± 0.008	0.979 ± 0.009	0.965 ± 0.010	22.3 ± 26.4
Tibia + Fibula R	0.975 ± 0.006	0.979 ± 0.006	0.972 ± 0.007	5.21 ± 4.59
Tibia + Fibula L	0.977 ± 0.002	0.981 ± 0.003	0.972 ± 0.004	4.19 ± 6.60

TABLE II: Metrics of bone segmentation predictions.



(a) Knee joint (b) Proximal femur (c) Pelvis

Fig. 2: Predictions of subject N14. Coloured based on the distance to ground truth.

there are some misclassified voxels, resulting in a higher Hausdorff distance (HD).

B. Novel data augmentation

1) Generalisability and sensitivity: Training, validating the model and generating 1000 shapes took 7h for the pelvis and 24h for the distal femur.

FlowSSM showed excellent generalisation on the distal femur dataset, indicating the successful construction of the latent space and capturing the underlying structure of the data. A CD of 0.177±0.03mm is found on the unseen cases. For the pelvis, CD values of 0.477 ± 0.04mm and HD values ranging from 2.869mm to 6.199mm are found. Despite being higher compared to the femur case, the mean CD of 0.477mm indicates that the model can still generalise effectively on unseen cases. The deformations with the smallest and largest HD showed that the model successfully matched the ground truths, with some areas exhibiting larger errors but overall reconstructing the shape accurately.

The study investigated whether FlowSSM could learn the orientation of shapes without any alignment preprocessing. The shapes were centred but not further aligned. Based on previous measurements of HD and CD and the visualisations that were investigated, it was inferred that the orientation did not pose a problem and was learned by the deformers.

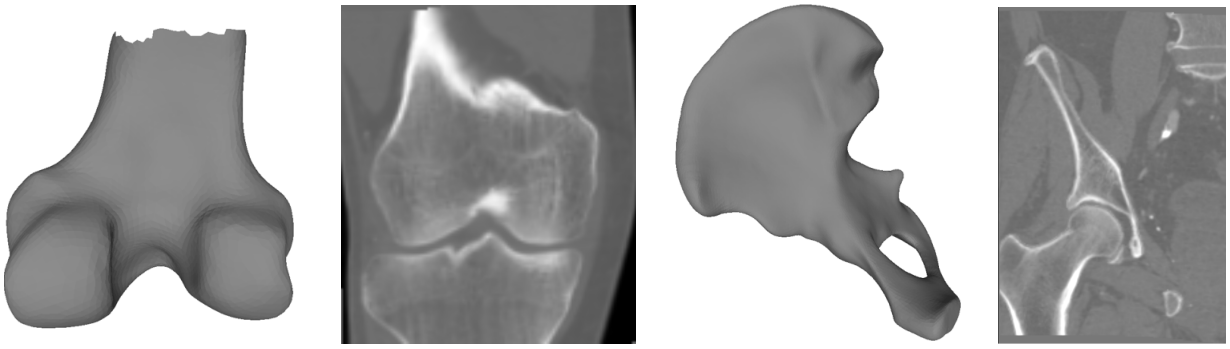
A specificity measurement of 0.287 ± 0.04 is obtained for the distal femur, and it can be compared to shapes generated by the DeepSSM PCA procedure. Using the DeepSSM framework, 1000 shapes are generated by fitting a KDE distribution to the PCA space, capturing the main modes and maintaining variability. The obtained specificity value for the PCA-generated shapes is 0.224 ± 0.01. This indicates that the model generating shapes from the PCA space is slightly more specific than the proposed model. However, since the values are within the same order, it can be concluded that the proposed model is capable of generating shapes that align with population statistics. Regarding the pelvis, a specificity measurement of 0.274 ± 0.20 is obtained, while the PCA method yields a specificity of 0.130 ± 0.01. Therefore, the PCA method appears to be more specific, especially for challenging cases like the pelvis. Nonetheless, the generated pelvis shapes visually appear acceptable.

2) TSP warp and results: Finding the TSP warp, to generate the images corresponding to the generated shapes, involves solving a linear system with equations based on control points. Careful selection of control points is important to balance processing time and accuracy. For the distal femur, 2000 random vertices out of 6151 were used, resulting in a good match. With 25967 vertices in the pelvis, more samples would be needed for accuracy, but it would take too long. Thus, 4800 vertices were chosen. Ideally, a faster method using all vertices would be preferred. Fig. 3 shows two examples of a generated shape, and the warped image. Note that it is confirmed that the generated images nicely match the generated shapes.

V. Discussion and Future Outlook

The field of medical image segmentation has increasingly adopted deep learning techniques for their speed and accuracy. Automatic bone segmentation in the lower limb region is particularly popular, but achieving a balance between speed and accuracy remains a challenge. Some studies focus on specific regions of interest (ROIs) but require manual cropping, limiting their clinical practicality. Others use whole-body CT scans but compromise on accuracy or speed.

Validation using a dataset of CT images from healthy patients demonstrates that the proposed workflow achieves faster predictions with comparable accuracy to existing methods. For example the methods Ambellan et al. and Kuiper et al.



(a) Generated shape (b) Generated image (c) Generated shape (d) Generated image

Fig. 3: Examples of generated shape and image pairs for both the distal femur and the pelvis.

achieve similar accuracy, but require respectively 2 hours and 20min of inference time on unseen cases [8], [12]. However, in the proposed method there are some issues such as leakage and misclassified voxels that need to be addressed for further improvements.

To enhance the workflow, refining the preprocessing workflow, optimising network architecture, and training procedures for each bone separately are recommended. Applying shape and appearance models as a postprocessing step could also improve accuracy, albeit with longer prediction times. Additionally, including additional bones and validating the workflow on diverse datasets are suggested.

A novel data augmentation method using FlowSSM for shape generation is introduced, which has proven effective in enhancing segmentation. However, improvements such as standardised hyperparameter tuning, discovering noncorrespondence-based warps, learning the distribution of latent embeddings, and exploring multiple objects or disconnected regions are suggested.

VI. Conclusion

This study presents a novel, fully automatic workflow for rapid and precise bone segmentation in whole-body CT scans of the lower limb. The workflow uses downsampled images to localise ROIs and then performs full-resolution segmentations. Next, a novel data augmentation method using FlowSSM for shape generation is introduced. Both the segmentation workflow and data augmentation technique show promise, but further improvements and adaptations are necessary to enhance their accuracy, reliability, and clinical significance.

References

[1] E. A. Audenaert, J. Van Houcke, D. F. Almeida, L. Paelinck *et al.*, "Cascaded statistical shape model based segmentation of the full lower limb in CT," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 22, no. 6, pp. 644–657, 2019. [Online]. Available: https://doi.org/10.1080/10255842.2019.1577828

[2] C. R. Henak, A. E. Anderson, and J. A. Weiss, "Subject-specific analysis of joint contact mechanics: Application to the study of osteoarthritis and surgical planning," *Journal of Biomechanical Engineering*, vol. 135, no. 2, pp. 1–26, 2013.

[3] M. Hans-Peter and T. Heimann, "Statistical shape models for 3d medical image segmentation: {A} review," *Medical Image Analysis*, vol. 13, pp. 543–563, 2009.

[4] C. Chu, C. Chen, L. Liu, and G. Zheng, "FACTS: Fully Automatic CT Segmentation of a Hip Joint," *Annals of Biomedical Engineering*, vol. 43, no. 5, pp. 1247–1259, 2015.

[5] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, no. December 2012, pp. 60–88, 2017.

[6] C. D. Naylor, "On the prospects for a (Deep) learning health care system," *JAMA - Journal of the American Medical Association*, vol. 320, no. 11, pp. 1099–1100, 2018.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015.

[8] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative," *Medical Image Analysis*, vol. 52, pp. 109–118, 2019. [Online]. Available: https://doi.org/10.1016/j.media.2018.11.009

[9] P. Liu, H. Han, Y. Du, H. Zhu *et al.*, "Deep learning to segment pelvic bones: large-scale CT datasets and baseline models," *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, no. 5, pp. 749–756, 2021. [Online]. Available: https://doi.org/10.1007/s11548-021-02363-8

[10] Y. Deng, L. Wang, C. Zhao, S. Tang *et al.*, "A deep learning-based approach to automatic proximal femur segmentation in quantitative CT images," *Medical and Biological Engineering and Computing*, vol. 60, no. 5, pp. 1417–1429, 2022. [Online]. Available: https://doi.org/10.1007/s11517-022-02529-9

[11] 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, no. 1, pp. 21–29, 2019. [Online]. Available: https://doi.org/10.1007/s11548-018-1883-7

[12] R. J. Kuiper, R. J. Sakkers, M. van Stralen, V. Arbabi *et al.*, "Efficient cascaded v-net optimization for lower extremity ct segmentation validated using bone morphology assessment," *Journal of Orthopaedic Research*, vol. 40, pp. 2894–2907, 12 2022.

[13] J. Cates, S. Elhabian, and R. Whitaker, "Shapeworks: particle-based shape correspondence and visualization software," in *Statistical Shape and Deformation Analysis*. Elsevier, 2017, pp. 257–298.

[14] R. Bhalodia, S. Elhabian, J. Adams, W. Tao *et al.*, "DeepSSM: A Blueprint for Image-to-Shape Deep Learning Models," 2021. [Online]. Available: http://arxiv.org/abs/2110.07152

[15] D. Lüdke, T. Amiranashvili, F. Ambellan, I. Ezhov *et al.*, "Landmark-free statistical shape modeling via neural flow deformations," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*. Springer Nature Switzerland, 2022, pp. 453–463.

[16] A. Adam, "Mesh voxelisation," 2023. [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation

Contents

List of figures

List of tables

Acronyms

2D - 2-dimensional.

3D - 3-dimensional.

AI - Artificial intelligence

ANN - Artificial neural network

ASM - Active shape models

Adam - Adaptive moment estimation

BCE - Binary cross entropy

CD - Chamfer distance

CE - Cross entropy

CNN - Convolutional neural network

CT - Computed tomography

DICOM - Digital Imaging and Communications in Medicine

DL - Deep learning

DSC - Dice similarity coefficient

FN - False negative

FP - False positive

GPA - Generalised procrustes analysis

GPU - Graphic processing unit

HD - Hausdorff distance

HU - Houndsfield units

KDE - Kernel density estimation

ML - Machine learning

MLP - Multilayer perceptron

OAI - Osteoarthritis Initiative

PCA - Principal component analysis

ReLu - Rectified linear unit

RBF - Radial basis function

ROI - Region of interest

SSM - Statistical shape models

TSP - Thin spline plate

TN - True negative

TP - True positive

UZ Gent - University Hospital of Ghent.

WCE - Weighted cross entropy

t-SNE - t-distributed stochastic neighbour embedding

1 Introduction

Advancements in orthopaedic and biomechanical research heavily rely on image segmentation of the bones in the lower limb region. However, the manual delineation of anatomical structures on medical images for segmentation is a time-consuming and labour-intensive process [1]. For instance, delineating the femur and pelvis alone can take up to 10 hours for a single case [2]. To overcome these limitations, there is a growing need for fully automated methods. While atlas-based methods and use of statistical shape models (SSM) that rely on prior knowledge have shown promising results [1, 3, 4], they still leave room for improvement in terms of segmentation accuracy and inference speed on new cases. For instance, the automatic segmentation of a full lower limb case using a statistical shape model method can still take up to 2 hours [1].

The use of deep learning (DL) for medical applications has seen a significant surge due to the increase in computing power, widespread availability of software, continuous improvements, and the abundance of data [5, 6]. Compared to more traditional techniques, DL techniques can offer fast and accurate predictions, making them advantageous. Particularly, convolutional neural networks (CNNs) have emerged as a powerful tool for interpreting medical imaging data, also for segmentation. The UNet architecture, introduced in 2015 by Ronnberger et al., has further propelled this trend towards using DL for medical image segmentation [7].

While recent DL research has yielded promising results in the area of bone segmentation, many studies focus on datasets that are centred around the region of interest (ROI) or have already been cropped to it [8–10]. Methods that operate on whole-body computed tomography (CT) scans often suffer from lower accuracy or slow inference times when applied to unseen cases [11, 12]. Moreover, these methods face additional challenges due to the variations in available data. Each network is trained and deployed on a distinct dataset, which can lead to issues when applying the model to different datasets. This makes it difficult to compare techniques directly. Hence, a standard method that achieves both accurate and fast predictions is still not readily available.

1.1 Objective and research methodology

The objective of this thesis is to develop a robust and accurate subject-specific segmentation workflow for the lower limb bones, which also provides fast predictions. Specifically, the workflow should have the ability to process a whole-body CT scan without requiring manual cropping to specific ROIs. This approach will be created and evaluated on an in-house dataset of whole-body CT scans of healthy patients, with the expectation that it will outperform the current state-of-the-art method in terms of both speed and accuracy. By integrating these ideas with upcoming research, a valuable tool can be developed that serves as the initial step for future orthopaedic studies that require subject-specific segmentations. In order to achieve this, the following research methodology is followed:

- The in-house dataset is thoroughly examined, and a well-defined preprocessing pipeline is established to ensure that the CT scans are properly prepared for the segmentation workflow.
- A processing strategy is devised that enables the whole-body CT to be processed in its entirety. To achieve this, the first step involves the development of a method that can accurately identify ROIs, which are the key bones of the lower limb: the pelvis, femur, tibia, and fibula.
- A CNN is developed and utilised to accurately detect the ROIs and generate sub-images for further analysis.
- A segmentation process is then applied to the sub-images using a CNN that has been specifically designed for each ROI. The resulting segmentations are then integrated to produce a final prediction of the entire whole-body CT.
- The ROI identification process, as well as the subsequent segmentation task, are both evaluated using appropriate metrics to ensure accuracy and reliability. The findings are then analysed and discussed in terms of their accuracy, speed, and clinical relevance.

One of the main challenges of many DL approaches is the lack of sufficient data for training. Therefore, alongside the segmentation method, a novel data augmentation technique is developed to address this issue. Although this augmentation method is not currently used in combination with the segmentation workflow, it may prove beneficial in future applications. The primary goal of this augmentation approach is to generate new shape and image pairs that conform to the modelled population. This is achieved through the following methodology:

- The available data is first preprocessed in a clear and precise manner to enable the implementation of the data augmentation technique.

- A novel method for generating new shapes is introduced and applied to the shape data to improve the currently used techniques.
- Further evaluation is conducted on both the shape generation method and the resulting generated shapes, with a focus on their generalisability and specificity. The specificity of the generated shapes is compared to an existing method to determine their relative effectiveness.
- The generated shapes are utilised to create new shape and image pairs, which can be used to train DL methods and improve their accuracy and robustness.
- The entire workflow is thoroughly discussed and evaluated to determine its efficacy, possible improvements, and potential applications.

1.2 Main contributions

The developed segmentation workflow addresses the primary challenges encountered in current segmentation methods, including those related to speed and accuracy when compared to both traditional and DL methods. Specifically, the following key contributions can be identified for the proposed segmentation workflow:

1. *Creating a workflow that can process whole-body CT scans in a fast and effective manner to find the ROIs.*

   Because most of the current deep learning research in medical imaging focuses on specific regions of the body using preprocessed images, there is a lack of standardised methods for processing whole-body CT scans. To overcome this issue, we have developed a novel method that uses a CNN to identify the major bones of the lower limb region - the femur, pelvis, tibia, and fibula - on a downsampled version of the original image. Our method provides a fast and reliable solution that can handle larger images, which current state-of-the-art methods often struggle to process. With this approach, we offer a simple yet effective solution that can improve upon existing methods for whole-body CT image processing.

2. *Use of the found ROIs to obtain accurate segmentations that compete with current state-of-the art.*

   The segmentation workflow not only offers a fast way to generate predictions but also produces accurate segmentations that can rival the current state-of-the-art methods. Our results are thoroughly evaluated, and we provide a clear research direction for possible future work. With our approach, we aim to contribute to the advancement of medical imaging segmentation, offering a reliable and efficient method for this important task.

The novel data augmentation method provides a way of generating new shape and image pairs, allowing future DL methods to benefit from it. For the shape generation, a technique called FlowSSM is used. The main contributions are:

3. *A thorough evaluation of the results of FlowSSM on two use cases.*

   As FlowSSM is a recently developed technique, only a few use cases have been explored so far. In this work, we evaluate its performance on two challenging cases: the distal femur and the complete pelvis. Through this thorough evaluation, we demonstrate that FlowSSM has the potential to be extended to more complex bone cases in the future. By showcasing its capabilities on these use cases, we aim to contribute to the development of FlowSSM as a promising technique.

4. *Evaluating the generated shapes by FlowSSM, showing no need for alignment.*

   Most existing shape generation methods require the shapes to be aligned as a preprocessing step, which can affect the generated images by enforcing a specific alignment. This alignment can, in turn, influence the performance of certain deep learning methods that are sensitive to image alignment. In this work, we explore the generalisability of our model without any alignment and the specificity of the generated shapes. To the best of our knowledge, this approach has not been previously investigated, as alignment was always applied before using FlowSSM. By exploring this option, we aim to contribute to the development of more flexible and robust shape generation methods.

1.3 Thesis outline

This thesis is organised as follows:

- *Chapter 2: Background and Related Work* presents the technical background that will be used to build and scaffold the theory in this thesis.
- *Chapter 3: Lower Limb Segmentation Method* outlines a DL workflow designed to segment the femur, pelvis, tibia, and fibula from a whole-body CT scan. The introduction of the general workflow is followed by a detailed exploration of two critical components, namely, bone localisation and bone segmentation. The available dataset and processing steps to complete this proposed method are given in more detail.
- *Chapter 4: Lower Limb Segmentation Results* presents and discusses the results of the proposed workflow. First, the metrics used to evaluate the results are given. Next, the results of both the bone localisation and segmentation components are presented, followed by a detailed discussion of these results.

- *Chapter 5: Neural-Flow Based Data Augmentation Method* introduces a novel data augmentation method that leverages FlowSSM, a technique that is explained in detail. The motivation behind the development of this method is first discussed. Finally, the necessary preprocessing steps required to apply the method to available data are elabourated upon.
- *Chapter 6: Neural-Flow Based Data Augmentation Results* discusses the results of the proposed data augmentation method and focuses mainly on the outcomes obtained with FlowSSM, with an emphasis on generalisability and specificity. Additionally, examples of the complete augmentation method are presented, and the method is discussed in more detail.
- *Chapter 7: Conclusion and Future Work* summarises the conducted research and obtained results. To improve our proposed methods, several future research directions are specified as well.

2 Background and related work

This chapter provides the essential background information necessary for a understanding of the methods employed in this thesis. A detailed explanation of the anatomical information of the bones examined in this thesis is provided, outlined in Section 2.1. Next, the concepts underlying computed tomography are elaborated on in Section 2.5. In Section 2.3, shape representation for working with 3D shapes is explained in more detail. Additionally, an in-depth explanation of deep learning is presented in Section 2.4, as the models developed in this thesis heavily rely on these principles. Finally, the objectives of segmentation are given, combined with an overview of the existing methods in Section 2.5.

2.1 Anatomy of the lower limb

The considered bones in this thesis are the pelvis, femur, tibia, and fibula. These are a subset of the bones in the lower limb. An understanding of the anatomy is necessary to fully grasp the results of this thesis. Each of these named bones is explained in more detail. For further detailed information, please refer to the chapter on the skeleton in Human Anatomy & Physiology of Marieb. E. and Hoehn. K. [13]. The anatomy being described pertains to healthy adults. This thesis specifically concentrates on the outer layer of bones. Therefore, a comprehensive analysis of the internal anatomy of bones is not included.

2.1.1 Pelvis

The pelvis is made up of two hip bones or coxal bones. Sometimes the sacrum is included too in the pelvis girdle. In this work, only the hip bones are referred to as the pelvis. These are connected by the pubic symphysis. Each of the hip bones consists of three major parts: the ilium, ischium, and pubis. These three are firmly fused in adults making them indistinguishable.

Together, they form the acetabulum, found on the lateral side, which is a cavity. The femur fits in this cavity making up the hip joint. Each hip bone joins the sacrum at the sacroiliac joint, making it the second joint of the hip bone. It should be noted that the pelvis is classified as a large, irregular bone. It is the only irregular bone considered in this thesis. Fig. 2.1 gives a more detailed look at the hip bones and their irregular shape.

The pelvis transmits body weight to the lower limbs, plays a vital role in childbirth as the child must pass through the birth canal, and supports the abdominal organs [14]. Because it plays a vital role in childbirth, there are clear differences between men and women. The female pelvis is tiled forward compared to those of men. Next, bone thickness is less for a woman and the acetabula are smaller and further apart.

2.1.2 Femur

The femur is the only bone found in the thigh and the first long bone considered in this thesis. It is the largest and strongest bone in the human body, having a length of roughly a fourth of a person's height. The femur has a proximal and distal extremity. Proximally, the head is found, which joins the hip in the acetabulum. The head has a ball-like shape, containing a small pit, which supports the attachment of the ligament that secures the femur to the acetabulum. The head goes over to the neck, the weakest part. Between the neck and the shaft of the bone, the greater and lesser trochanters can be found, which serve as attachment sites for muscles.

The distal part of the femur on the other hand forms the first part of the knee joint. Here the lateral and medial condyles can be distinguished. The femur can be seen in more detail in Fig. 2.2.

2.1.3 Tibia and fibula

The tibia and fibula are two parallel bones that form the skeleton between the knee and ankle. They articulate both distally and proximally.

The tibia is the second bone in the human body after the femur in terms of size and strength. It transmits the weight of the body from the femur to the foot. The proximal end contains concave medial and lateral condyles. Between these, the intercondylar eminence is found. The condyles articulate with the corresponding ones from the femur. Distally, the tibia is flat articulating with the foot. Here the medial malleolus is found, which forms the medial bulge of the ankle. Together with the femur and patella, the tibia forms the knee joint. The patella can also be referred to as the knee cap but is not further investigated in this thesis.

The fibula is a long bone and ends proximally in the head of the fibula. Distally it ends in the lateral malleolus, which forms the lateral ankle bulge that articulates with the talus. The function of the fibula is not to carry weight, but several muscles originate from it. More details on both the tibia and fibula can be found in Fig. 2.3.

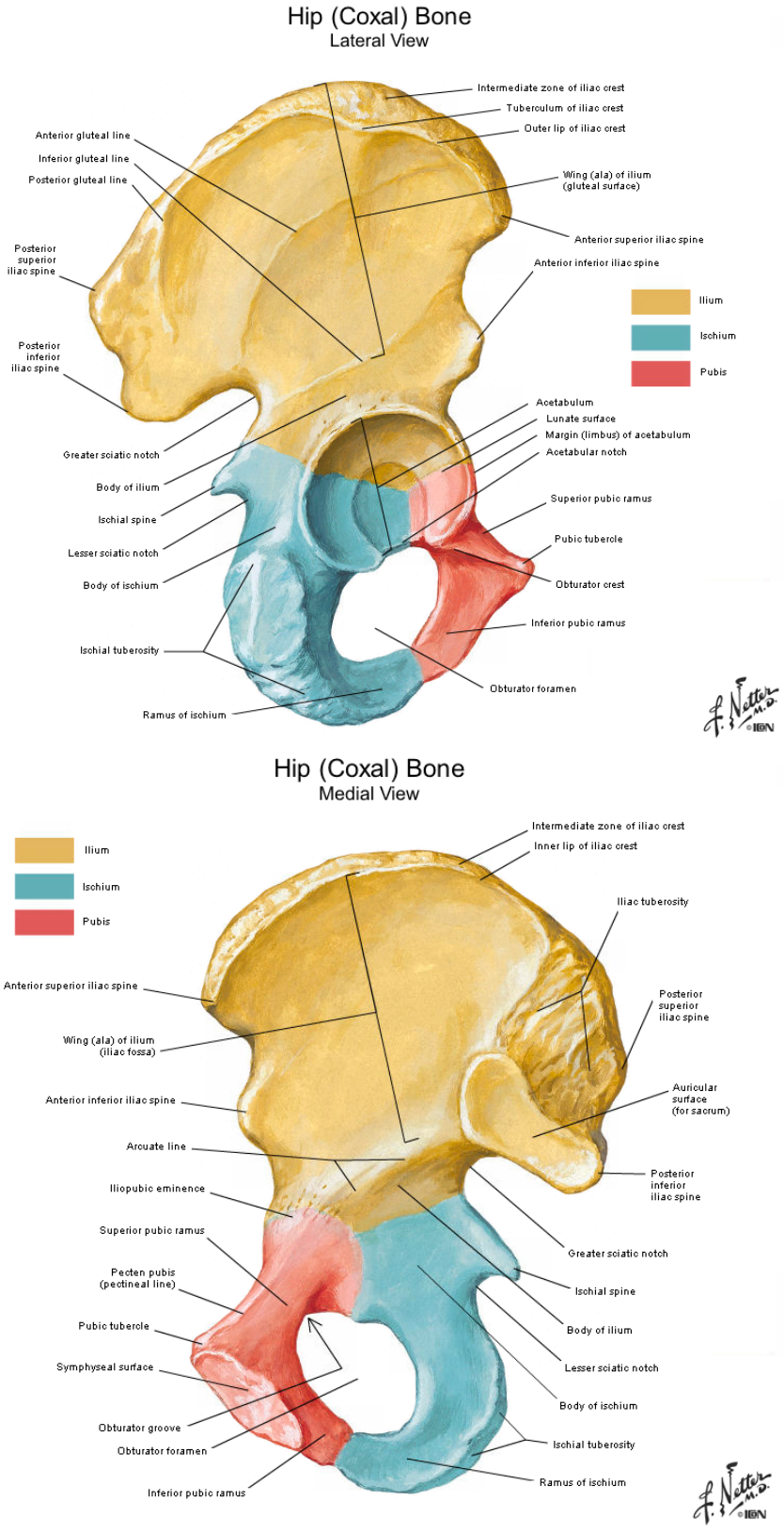


Figure 2.1: Anatomy of the pelvis [15].

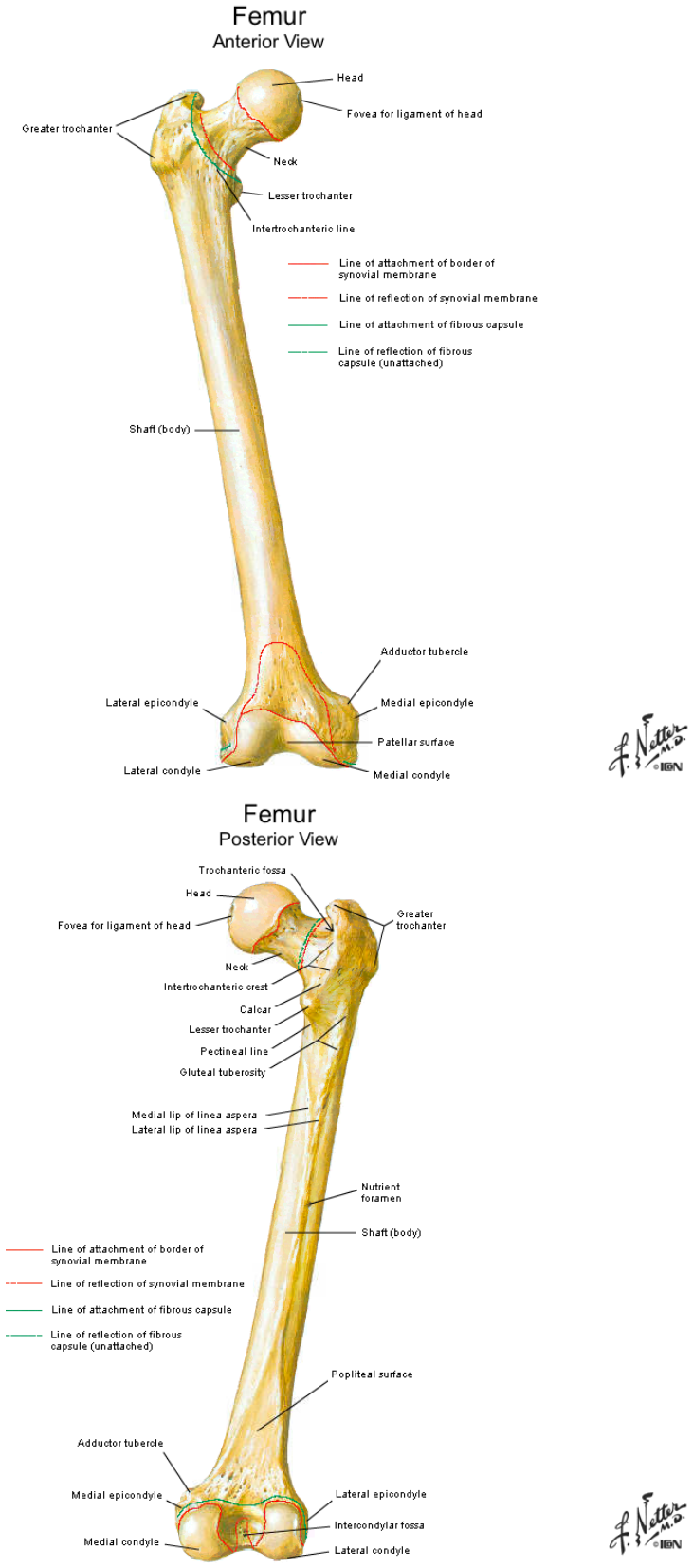


Figure 2.2: Anatomy of the femur [15].

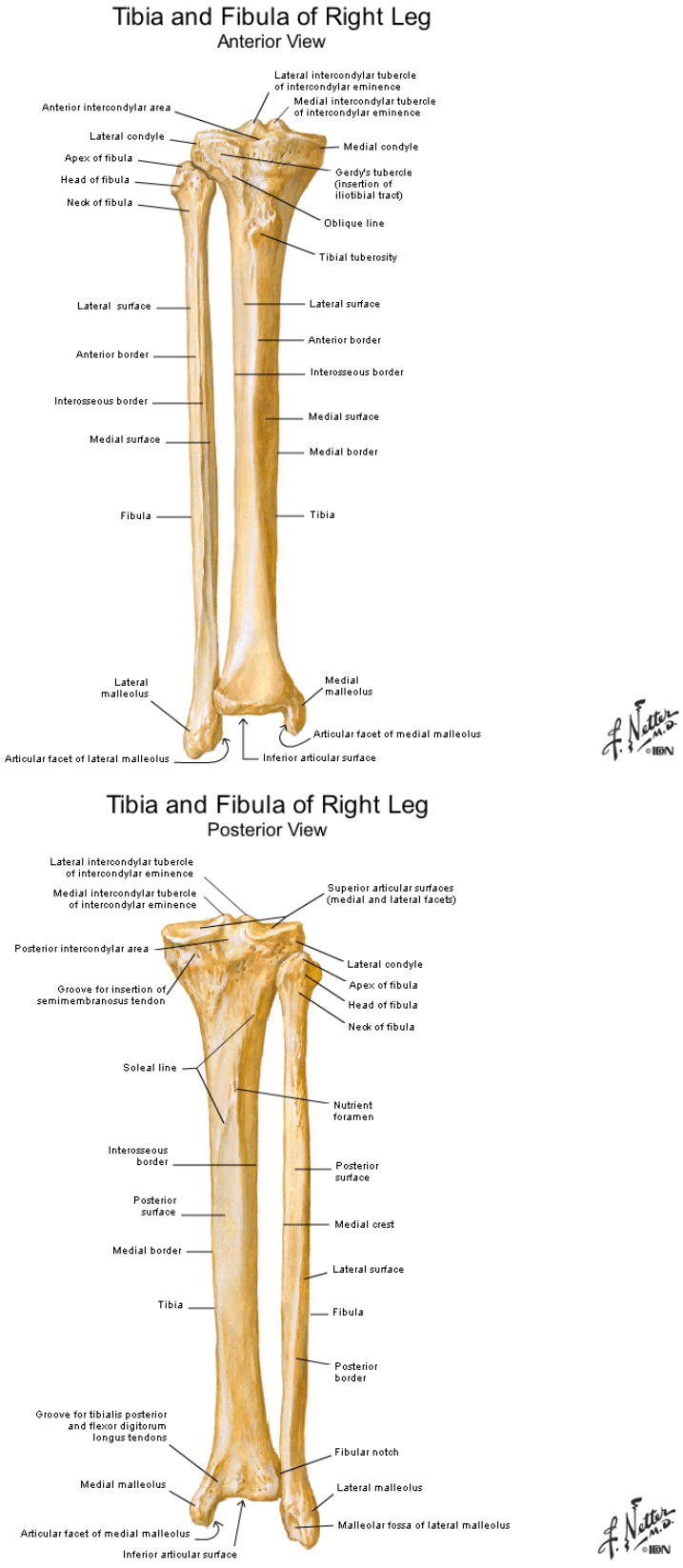


Figure 2.3: Anatomy of the tibia and fibula [15].

2.2 Computed tomography

For the study of the lower limb bones, a noninvasive, imaging technique called computed tomography (CT) is used. It combines X-rays with computer processing to obtain a detailed 3-dimensional (3D) view of the internal structures of the human body. CT uses an X-ray source combined with an X-ray detector that rotates around the patient. The detectors measure the intensity of the X-ray that has passed through the patient. Visualisation of different structures of the human body is possible because the intensity of the photons in the X-rays is reduced based on the tissue-specific attenuation coefficient μ. The attenuation coefficient is a measurement of how much a certain tissue scatters or absorbs the radiation passed through it. Different tissues have different values of μ. The attenuation of bones is for example much larger than that of soft tissue, making them appear white on a CT. The intensity that the X-ray detector receives can be defined by:

$$I = \int I_0(E) \exp\left(\sum_i -\mu(E)_i x_i\right) \mathrm{d}E, \tag{2.1}$$

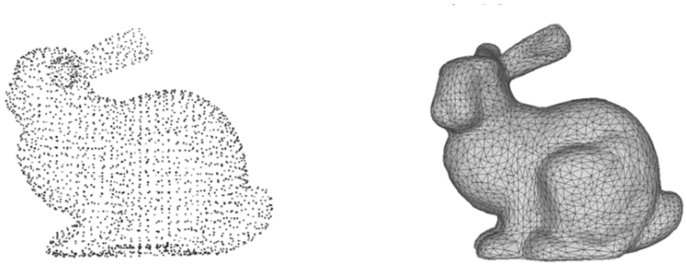
where $I_0(E)$ is the intensity of the X-ray given by the source. The measured intensity depends on the summation of the linear attenuation coefficient μ_i of all materials i times its linear extent x_i. I_0 and μ_i are dependent on the photon energy E. Hence, an integration over the total X-ray energy spectrum gives is needed.

A CT scan acquires these intensities at different discrete angles between 0° and 360°. After the data collection, a reconstruction algorithm is used to calculate cross-sections of the human body. The output of the reconstruction is a voxel-by-voxel map of the patient. A voxel is a 3D pixel representing the smallest unit of data in the obtained 3D image. The smaller the voxel size, the higher the resolution of the CT scan. The values of the voxels indicate the beam attenuation of the underlying structure. These values are scaled to Houndsfield Units (HU), which indicates the ratio of the linear attenuation coefficient of the voxel to that of water. This can be written as:

$$CT(x, y) = 1000 \frac{\mu(x, y) - \mu_{water}}{\mu_{water}}. \tag{2.2}$$

Structures with a higher HU appear bright in the image, while images with a lower HU appear dark. While the HU value of soft tissue ranges from 100-300HU, that of bones ranges from 300-2000HU making bones appear bright on CT scans.

The advantage of CT is that it is readily available at most hospitals, providing a fast and high-resolution view. It is seen as one of the most reliable techniques in bone assessment [16]. On the other hand, CT exposes the patient and staff to radiation.



(a) Point cloud (b) Mesh

Figure 2.4: Example of a point cloud and a mesh [23].

2.3 3D shape representation and correspondence

The use of 3D models in the medical field gained more popularity and interest due to the availability of novel machines, softwares etc. [17, 18]. 3D models can be used for all kinds of different applications, ranging from 3D models for pre-operative planning to implanting 3D-printed patient-specific models during surgery [19, 20]. Therefore, the construction of a digital representation of a 3D shape is an important task. In this thesis, the definition of Lamecker is followed, stating that a shape refers to the geometric information represented by the boundary of a 3D object $X \subset \mathbb{R}^3$ [21]. Note that this boundary X can be represented by different kinds of digital representations. In this thesis, shapes are represented in two ways, which are used interchangeably. One of the simplest ways of representing a shape is a point cloud, a dense set of 3D points. A point cloud P in 3D containing n points can be denoted by $P = \{(x_1, y_1, z_1), ..., (x_n, y_n, z_n)\}$. Another way of representing a shape is by making use of meshes. Meshes can be seen as an extension of a point cloud. They exist out of vertices, edges and faces, which together form the shape. The vertices are a set of 3D points, just like the point cloud. Vertices are connected by edges and faces form the surface. It is possible to convert a mesh to a point cloud by only considering the vertices. Point clouds can be converted to meshes by utilising certain algorithms, e.g. the ball pivoting algorithm is used for creating a triangular meshes by interpolating point clouds [22]. Fig. 2.4 shows an example of both a point cloud and a mesh.

An important feature of shapes used in this thesis is correspondence. A shape correspondence problem can be explained by the following statement: given N input shapes $X_1, X_2, ..., X_N$, establish a meaningful relation $\mathcal{R}$ between their elements. Two shapes are in correspondence when $(s, z) \in \mathcal{R}$ for elements $s \in X_i$ and $z \in X_j$ with $i \neq j$ [24]. One-to-one, one-to-many or many-to-many correspondences exist. In this thesis, when referring to shapes that are in correspondence with each other, a many-to-many relation is assumed, having a full relation between all shapes. Establishing correspondence is often a fundamental operation in all kinds of applications like shape interpolation, shape reconstruction, statistical shape modeling, etc. [25].

In the case of medical shapes, correspondence can be manually established by experts by indicating points on different shapes referring to the same anatomical locations. These points are often referred to as landmarks. However, this is a complex and tedious, time consuming, task, especially when one needs to obtain a dense set of correspondence points. Next, using a manual method is prone to mistakes due to inter- and intra-observer variances [26]. Hence (semi-)automatic methods are used to obtain a dense set of correspondences. An overview of these techniques can be found in the work of Sahillioglu [25].

2.4 Deep learning

To effectively comprehend the models employed in this thesis, it is essential to have a grasp of the principles that form the basis of Deep Learning (DL). This section provides the necessary information to have a comprehensive understanding of the constructed models. For additional information, please refer to Goodfellow et al.'s book [27].

DL is a specialised branch of Machine Learning (ML), which is itself a subset of Artificial Intelligence (AI). AI refers to systems that can demonstrate intelligent behaviour and characteristics, such as reasoning, learning, perception, and decision-making [28]. ML, on the other hand, refers to the ability to acquire patterns from raw data. When this ability to learn and improve from vast amounts of data is achieved through the use of multiple layers of complex transformations, it is known as DL. Over the past few years, the architecture of DL models has been evolving to include an increasing number of layers, which has led to significantly improved performance in various applications. This trend towards deeper models is primarily driven by the increasing availability of data and advancements in computational power. Mainly the increasing power and affordability of Graphics Processing Units (GPUs) have played a major role in driving forward the faster and more cost-effective performance of calculations [29].

This thesis employs one of the fundamental types of deep learning known as supervised learning. This approach involves the use of a labelled dataset, which contains both inputs and corresponding outputs. The main goal of supervised learning in deep learning is to learn how to map input data to desired output data by minimising the difference between predicted and actual outputs. In medical imaging techniques, desired outputs can take various forms, such as a single label for an entire image or individual labels for each voxel.

The subsequent sections provide a more detailed explanation of the essential building blocks required to comprehend a DL model and its learning process.

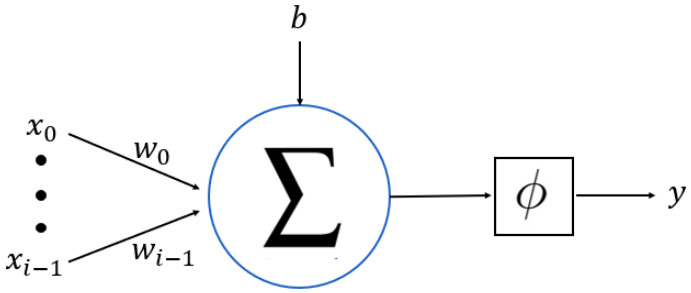


Figure 2.5: Perceptron with i inputs.

2.4.1 Artificial neural networks

To extract meaningful insights from data, deep learning models are designed to simulate the cognitive processes of the human brain. To understand this process, it is essential to first comprehend the underlying principles of an artificial neural network (ANN), which serves as the foundation for DL. By exploring the inner workings of ANN and its ability to learn, a better understanding of the key concepts and techniques that are essential for DL is gained.

ANNs are made up of perceptrons, which are modelled after neurons [30]. Thus, the architecture of an ANN is designed to mimic the structure and function of the human brain. An example of a perceptron is given in Fig. 2.5. The perceptron takes $x_0, ..., x_{i-1}$ as inputs and multiplies it with the corresponding weights $w_0, ..., w_{i-1}$. These are summed together with bias b. The output y is found after applying an activation function ϕ to the summation. These operations can be formulated as:

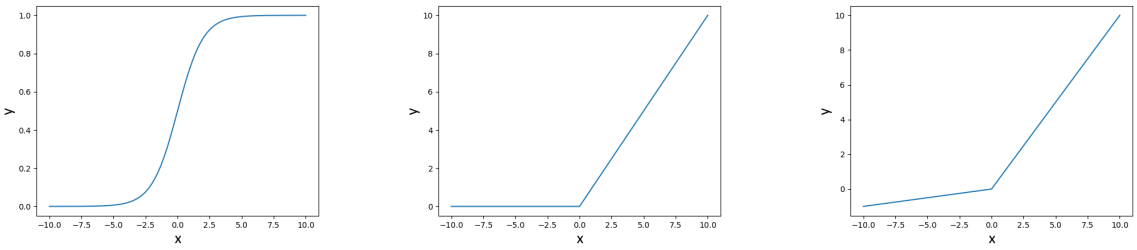
$$y = \phi\left(\sum_{i=0}^{i-1} w_i x_i + b\right). \tag{2.3}$$

The activation function ϕ determines the final output of the perceptron. To capture nonlinear relationships between input and output, it is essential to use a nonlinear activation function. Different kind of activation functions exist. Some of the popular functions are the sigmoid, rectified linear unit (ReLu) and leaky ReLu. These are found in Fig. 2.6 for values of x between -10 and 10. In this thesis, the ReLu activation is mostly used, defined for an input value x as:

$$\text{ReLu}(x) = \max(0, x). \tag{2.4}$$

Figure 2.7 illustrates an example of an artificial neural network (ANN) architecture built upon these perceptrons. The network consists of one input layer with d neurons, M hidden layers, and one output layer with m neurons.

As stated before, the goal of training an ANN for a certain task is to learn how to map the input data to desired output data by minimising the difference between predicted and actual outputs.



(a) Sigmoid (b) ReLu (c) Leaky ReLu

Figure 2.6: Different activation functions.

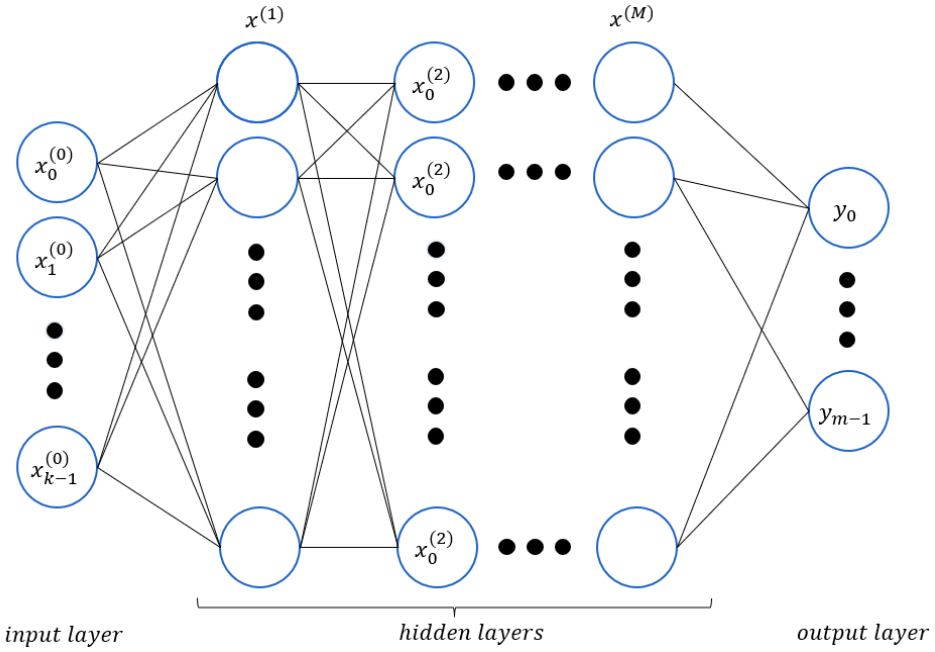


Figure 2.7: ANN architecture with M hidden layers.

This can be done by finding the optimal weights and biases that minimise this difference. To quantify this difference, a loss function is used. A loss function measures the difference between the predicted and desired output data. Different kinds of loss functions are proposed and are often specific to the certain task at hand. An important feature of a loss function is that it gets smaller when predictions improve.

The training of an ANN has three major parts and is mostly based on a gradient descent method. First, the input data is passed forward through the network and the loss of the outputs is calculated. Next, the gradients of the loss function w.r.t. the weights and biases are calculated using the chain rule. This is called backpropagation [31]. Finally, weights and biases are updated by multiplying these gradients with a learning rate and subtracting them from the network parameters. A parameter p_j is updated using:

$$(p_j)_{k+1} = (p_j)_k - \alpha \frac{\delta L}{\delta p_j}, \tag{2.5}$$

with k an iteration step, α the learning rate and L the loss function of the considered training samples. Parameters can be updated after a single training sample, or after receiving a larger part of the dataset, named a batch. Updates after a single training sample might lead to a faster convergence of the loss function to a minimum, but can also cause fluctuations. Hence, often a batch is preferred [32].

Note that the training does not only depend on the loss function. The optimisation method and hyperparameters of the training procedure are important too. Currently, the adaptive moment estimation (Adam), a first-order gradient-based method, is one of the most widely used optimisers [33]. It remains a viable option for a lot of applications [34]. Hyperparameters include the batch size, the learning rate, and the number of epochs. An epoch is one complete cycle through the entire training dataset during the training.

By setting the appropriate loss function, optimiser and hyperparameters, the goal is to update the parameters of the network so that the network is able to generalise well enough to deal with cases that are not used for training.

2.4.2 Convolutional neural networks

A popular type of neural network is the convolution neural network (CNN). CNNs are mainly used in the field of pattern recognition within images. The main benefit is the reduction of parameters compared to a standard ANN [35]. A fully connected network with a single neuron in the first layer requires already 748 weights for a small 2-dimensional (2D) image of 28x28 pixels. Hence, applying a standard ANN on larger images, including more hidden layers would be an unmanageable task.

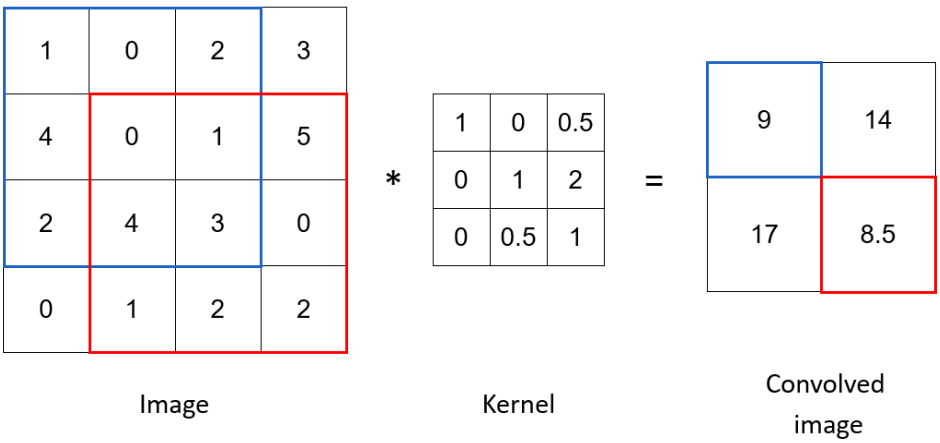


Figure 2.8: Example of a convolution.

The input of a CNN is a 2D image of pixels or a 3D image of voxels. Multilayered inputs are also possible. Instead of the typical perceptrons used in ANNs, CNNs make use of convolutional layers, which apply several filters or kernels to extract features. Kernels are either 2D or 3D. A kernel in a certain convolutional layer is convoluted with the result of the previous layer. This means that the kernel slides over an image and returns the weighted sum of its own weights and the image values. Applying a kernel K with size (m, n) to a 2D image I gives

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n). \tag{2.6}$$

found at position (i, j) of S, the resulting image after convolution [27]. An example of a convolution operation for an input image of size (4,4) and a kernel size of 3 is given in Fig. 2.8. Note that the size of the resulting image after convolution depends on the step size of the kernel, also referred to as stride. In Fig. 2.8 a stride of 1 in both directions is used. Next to the kernel size and the stride, input images before convolution are often padded to make sure that the output size remains equal to the input size. Examples of existing padding methods are zero-padding, circular padding, and reflection padding [36].

By stacking multiple of these convolutional layers, one can build a CNN. Each convolutional layer is often followed by an activation function to introduce nonlinearities. This is similar to an ANN. Next, another operation that is often applied after a convolutional layer is called pooling. Pooling is an operation that downsamples the resulting image. The operation is based on sliding a kernel with a size over the image with a certain stride. The stride is taken larger than 1 to reduce the output size. Common pooling operations are max pooling, average pooling, and L_p pooling [37]. In this thesis, max pooling is used. Max pooling takes the element of maximum value within the investigated neighbourhood. Next to downsampling, images can also be upsampled, often done by applying transposed convolutions. They are essentially the inverse of regular convolutions, where the input size is enlarged instead of compressed.

By combining the explained operations of convolution, activation functions and pooling, one can build a CNN with a specific architecture for the task at hand. In a classification task, fully connected layers are typically included to interpret the output of the convolutional layers and produce the class probabilities. However, in a segmentation task, fully connected layers may not be required.

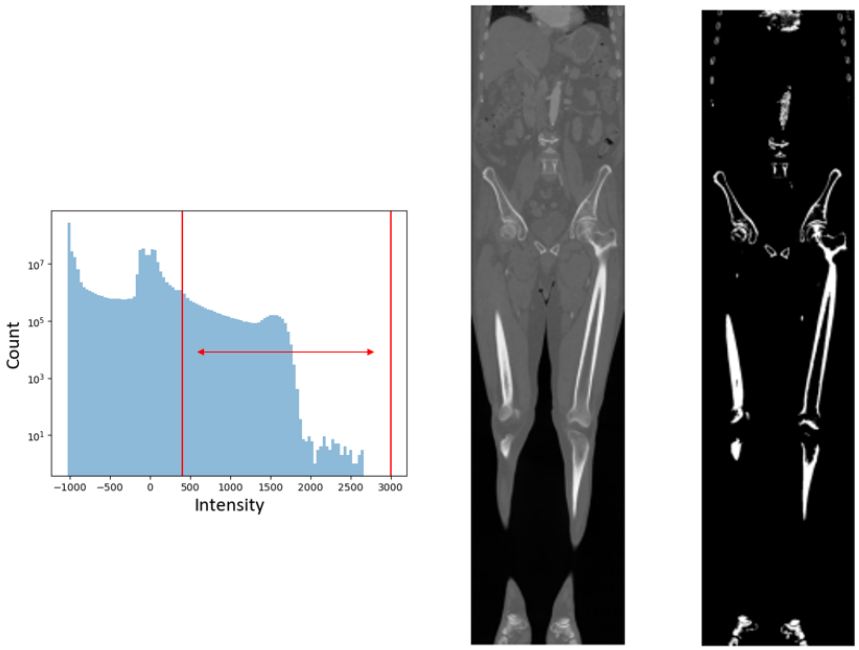
2.5 Segmentation

2.5.1 Segmentation goal and difficulties

Segmentation is the process of dividing an image into nonoverlapping regions that share certain characteristics, such as intensity or texture, to create a meaningful representation of the image [38]. In medical imaging, segmentation is used to delineate anatomical structures of interest, enabling further analysis. Further processing of these medical images, like feature extraction or dynamic simulations often largely depends on the accuracy of the segmentation. Since medical images are represented as digital maps of pixels or voxels, semantic segmentation is a typically used kind of segmentation, where each pixel or voxel in the image is labelled with a specific class. Common classes in medical images include the background and anatomical structures.

Manual delineation of anatomical structures by experts is the golden standard for segmentation on medical images. However, this approach is time-consuming and labour-intensive, which limits its practicality for large-scale research or rapid clinical results for patients [39]. For instance, it can take experts up to 20 minutes to segment a single bone class in lower limb CT scans [12]. Next, manual segmentation is prone to interoperator variability, definitely for certain pathological cases [40]. To address these issues, (semi-)automatic segmentation methods have been developed that do not require manual input. While there are general methods available, specialised approaches may be necessary for specific applications to achieve higher accuracy. Nonetheless, there is a wide discrepancy in imaging modalities and segmentation requirements, and no single method has been widely adopted as the standard for yielding optimal results.

In the case of bones, the main challenge lies in the segmentation of the joint regions, because of the close spatial relation [1]. The irregular shape of certain bones can make automatic segmentation a difficult challenge, definitely in joint epiphysis areas, due to the less pronounced contrast with soft tissue and the narrow inter-bone space [41]. The following parts of this section explain proposed methods for bone segmentation in the lower limbs and explain the difficulties that each method faces. First traditional segmentation methods are introduced. With traditional, the use of no AI is meant. Secondly, DL techniques for bone segmentation are introduced.



(a) Intensity histogram (b) CT (c) Mask

Figure 2.9: Example of thresholding applied to CT.

2.5.2 Traditional automatic segmentation methods

Thresholding

Thresholding is one of the simplest ways to obtain fast and automatic segmentations of the bones in the lower limb region. The goal is to determine certain intensity values, the thresholds, to separate the desired classes. Segmentations are obtained by grouping pixels with intensities within the same range, separated by the threshold, together. This has been widely applied to medical images [42]. Intensity values in the case of CT are determined by the HU of the voxels. Different kinds of tissues are within different ranges of intensity values, allowing setting certain thresholds.

Determining the threshold values is an important task to obtain good results. Nevertheless, it is not an easy task. Thresholding techniques can be divided into two major groups: global and local thresholding. In global thresholding, a manual or automatic threshold is set for the full image. Each pixel or voxel is evaluated separately and compared with the preset value. A simple, not optimised, example of global thresholding applied to a CT image can be seen in Fig. 2.9. Every voxel within a preset range of 400-3000HU is included in the mask. This example illustrates the challenge of identifying the appropriate range of intensity values that avoid both over- and undersegmentation of the bones.

The values in the example are chosen manually. However, automatically found preset values are possible. One of the most famous algorithms that automatically finds a threshold is Otsu's method. Here the threshold is found by minimising intraclass intensity variance [43]. While global thresholding is a fast and easy method, it can fail in CT images due to the partial volume effect, beam hardening, intensity inhomogeneity of bone structures, and high grey level of surrounding pixels [44]. Therefore, local thresholding methods have been proposed, calculating a different threshold for each pixel or voxel based on the neighbourhood statistics.

However, none of these techniques have proven to be fast and accurate enough in the case of bone segmentations of the lower limbs. Thresholding methods are susceptible to under- or oversegmentation based on the thresholds. Due to diffused boundaries, it is almost impossible to find a threshold that fully includes the bones, without including other tissues [44].

Statistical shape models

Statistical shape models (SSM) can be applied to obtain personalised segmentations while working with a small dataset, which is often the main difficulty when working with deep learning techniques, explained further [1]. The technique constructs an anatomical SSM based on a training set, which statistically describes a shape consistent across the population. The SSM consists of the mean shape and the main modes of variation. After this step, the SSM can be fitted to new images to get personalised segmentations. An optional final step is to apply local refinements as explicit local differences are harder to model and fit [26].

The first step in applying SSM to shape data of the population is by defining the kind of shape representation. A classical approach is placing landmarks on specific anatomic locations. More recently, a dense set of correspondences is being used. These are a set of consistent particles placed on surfaces. Shape correspondence is needed for the SSM to be consistent.

Let $\mathbf{X_i} = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n}\}$ be a point distribution model of a 3D shape, where each $x_{i,j} \in \mathbb{R}^3$ with $i = 1, \ldots, M$ and $j = 1, \ldots, n$, is a 3D coordinate represented as a vector of real numbers. Having these point cloud representations for each shape allows for building the shape model.

Before building the model itself, the shapes need to be aligned. This can be achieved by using different techniques. A popular technique is the use of generalised procrustes analysis (GPA) [45]. Once alignment is achieved, the mean shape and modes of variation can be calculated. The mean shape $\bar{\mathbf{X}}$ of a total of M shapes $\mathbf{X}_i$ can be easily obtained using:

$$\bar{\mathbf{X}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{X}_i. \tag{2.7}$$

The modes of variation, also called principal components, are found by applying principal com-

ponent analysis (PCA). These are computed from the covariance matrix C found by:

$$C = \frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{X_i} - \bar{\mathbf{X}})^T(\mathbf{X_i} - \bar{\mathbf{X}}). \tag{2.8}$$

An eigendecomposition of C, done by solving

$$C\mathbf{v_i} = \lambda_i \mathbf{v_i}, \tag{2.9}$$

gives the modes of variation $\mathbf{v} = \{\mathbf{v_1}, ..., \mathbf{v_{M-1}}\}$ and their variances $\lambda = \{\lambda_1, ..., \lambda_{M-1}\}$. The modes are sorted based on descending value of λ, so that $\lambda_1 > \lambda_2 > ... > \lambda_{M-1}$.

It is now possible to represent each shape by a linear combination of these modes of variation, using:

$$\mathbf{X_i} = \bar{\mathbf{X}} + \sum_{t=1}^{c} b_t \mathbf{v_t}. \tag{2.10}$$

The number of used modes c is chosen so that the summation of λ_1 till λ_c reaches a certain preset value, often 0.9-0.95. In this way, applying PCA allows representing each shape by a shape-specific latent representation $\mathbf{b} = \{b_1, ..., b_c\}$.

Once a shape model is built, it can be fit onto unseen images. Different techniques can be used to complete this final step. A popular technique is the use of active shape models (ASM). Note that SSM is a very useful technique that has been very accurate in a lot of cases. In addition to this, the technique can be seen as a white box, as each processing step can be clearly explained. A disadvantage is that fitting the model to new cases can require a large calculation time. Next to this, fitting aberrant shapes, like unexpected tumours or bone fractures, remains difficult. Therefore, in certain subjects with specific pathologies, SSMs can be harder to use for segmentation.

2.5.3 Deep learning segmentation methods

The use of deep learning has significantly increased over the past years in all fields. This is also the case for the task of automatic segmentation from medical images. Hence, two of the possible methods to obtain segmentation using DL are presented.

UNet

A neural network that has quickly become the golden standard for segmentation on medical images using DL is the UNet, developed in 2015 for biomedical segmentation [7]. UNet is a CNN that can be used for both 2D and 3D images. The goal of the network is to classify each pixel or voxel of a digited image, either as an anatomical structure or as a background. The UNet, named after its symmetrical shape, consists of an encoding path where higher-order

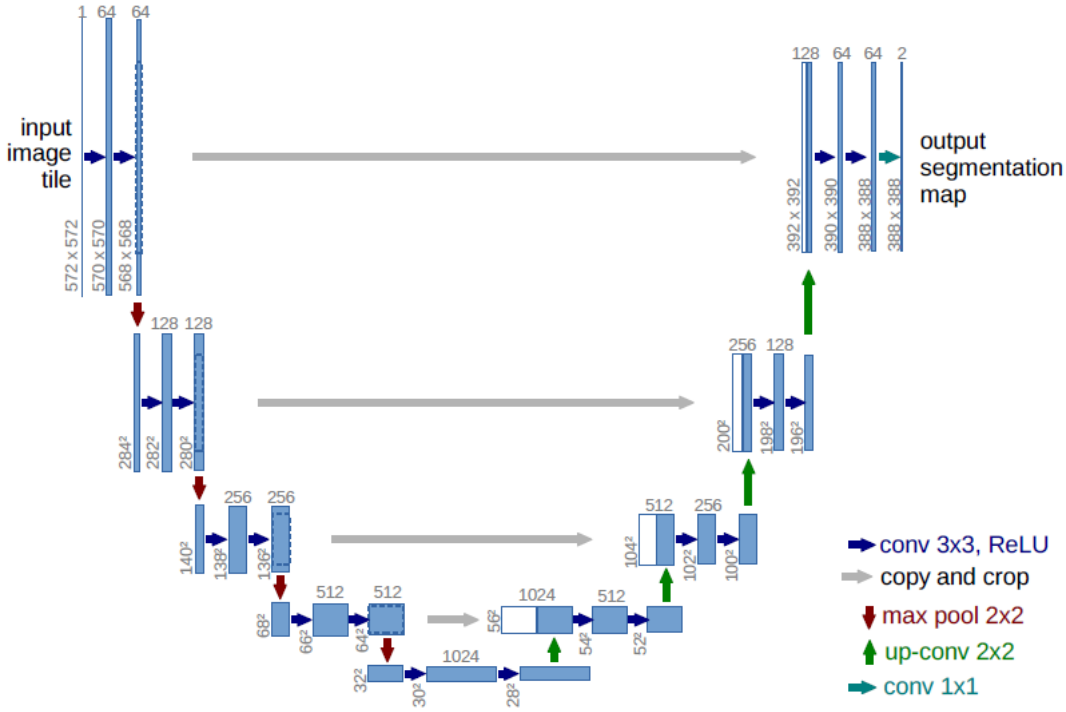


Figure 2.10: Original UNet as proposed in 2015 [7].

features are captured in downsampled resolutions, followed by an expanding path to translate these features to the original input size. This is combined with skip connections, which apply concatenations between the encoding and decoding paths. Thanks to these skip connections, context is captured from both smaller and larger features, giving good localisation. This gives the ability to do pixel- or voxel-level segmentation. The original architecture can be found in Fig. 2.10. The network is very powerful and can be applied to all kinds of segmentation tasks, including multi-class segmentation. Combined with its accuracy and possibility for real-time predictions this has resulted in heavy adoption and creation of variants of this network for medical image segmentation [46].

Applying UNet for medical image segmentation gives rise to several challenges. The first one is the availability of computation power. The network can be computationally heavy to train and deploy. This is definitely the case for larger images. This means that for example full-body CTs cannot be processed as a whole. As is the case for any deep learning application, the amount of labelled data to obtain good results can be large, depending on the task at hand. In the case of medical image segmentation, it is difficult to have large datasets as annotating images is labourious work. Hence, if the amount of available data is too scarce, it becomes difficult to obtain a segmentation model that can generalise well on new unseen imaging data. However, UNet showed to be very powerful in cases where even the data amount is too small for other DL techniques. Next, images of other modalities, or even different imaging settings within the

same modality, can be difficult to predict using a trained UNet on a different dataset. Because no prior anatomical information is included when training the network, it is possible that the UNet predicts things that are clearly anatomically impossible. Therefore, combinations of UNet with other techniques like SSM are being developed to include anatomical information [8]. A final difficulty is achieving a high-quality 3D surface, this is due to the pixel- or voxel-wise segmentation. To obtain smooth surfaces, postprocessing steps are required.

DeepSSM

A more recent idea to obtain 3D models of anatomical structures is called DeepSSM, developed by ShapeWorks [47, 48]. The goal to combine the strengths of SSM and deep learning. DeepSSM circumvents a heavy preprocessing and segmentation pipeline needed in classical SSM. It gives the possibility to achieve fast functional mappings from images to a low-dimensional shape descriptor based on deep learning. This is obtained by training a CNN that is trained to directly predict these shape descriptors from images, which can then be converted into 3D shapes. Inference on new samples is computationally fast when comparing it to state-of-the-art classical SSM approaches. The used shape predictors in their application are the PCA latent representations $\mathbf{b}$.

In order to build a prominent CNN model to directly predict low-dimensional shape descriptors from images, one needs enough representative training data. The need for data for this application is higher than for a UNet. Therefore, the introduction of DeepSSM is accompanied by a new data augmentation method to increase training size. This method is explained in Section 5.2. With DeepSSM, Shapeworks provides a blueprint for direct shape prediction from medical images. This can be beneficial compared to techniques like the UNet, which produce a labelled map of the structures. These need to be processed to obtain the 3D models themselves. However, this technique is still a work in progress. In this thesis, the focus is on creating the data augmentation technique.

2.6 Conclusion

This chapter introduced the needed information to fully understand the work proposed in the following chapters. The necessary understanding of the bone anatomy in the lower limb is given. Next, the technical background of deep learning is explained, followed by an explanation of image segmentation and an understanding of how deep learning can help. In the upcoming chapters, these techniques will be employed to develop a segmentation tool that leverages the predictive power of deep learning, in comparison to other methods such as the use of SSM.

3 Lower limb bone segmentation method

In this chapter, a workflow is proposed to generate precise subject-specific 3D models of the lower limbs from a full CT image. The primary objective of this workflow is to minimise the required calculation time while ensuring high accuracy. The workflow has the ability to process a whole-body CT scan without requiring manual cropping to specific ROIs. This approach effectively resolves the challenges associated with state-of-the-art segmentation methods that utilise SSMs. Specifically, the time-consuming nature of the calculation process can be mitigated. While DL techniques could potentially assist with fitting to complex bone fractures, this thesis does not focus on these issues as they are not represented in the available data. With this approach, a clinician can efficiently use the workflow by providing a full CT scan as input and receiving the 3D models promptly. The proposed task is accomplished by leveraging the advantages of UNet, such as its multiclass prediction capability, fast processing, and flexibility for diverse segmentation problems. In addition, the proposed workflow addresses the challenge of limited computing resources, making it easier to execute the segmentation process efficiently. This chapter explains the proposed workflow in further detail. Section 3.1 provides an overview of the used dataset. Moving to Section 3.2, the workflow of this project is presented. Sections 3.3 and 3.4 focus on the specifics of two critical steps in the workflow, namely, bone localisation and segmentation. In Section 3.5, the metrics used for evaluating the models are explained.

3.1 Dataset

DL requires a significant amount of annotated data to achieve optimal performance and generalise well to new cases. Therefore, having a robust dataset is essential to train, deploy, and test deep learning models accurately, particularly when dealing with accurate segmentation. In this thesis, a dataset comprising information from 94 healthy subjects, each of whom has an available full-body CT image, is utilised. The medical data is stored as a series of Digital Imaging and

Bone Type	Number of Vertices	Number of Faces	Edge Length
Femur	21 096	42 188	1.77 ± 0.31
Pelvis	25 967	51 934	1.57 ± 0.27
Tibia and Fibula	39 197	78 386	1.14 ± 0.32

Table 3.1: Number of vertices and faces of the outer bone layer.

Communications in Medicine (DICOM) files, which is the standard format for medical imaging.

The DICOM file type allows reading the metadata of each image, providing additional information about the population and scanning procedure. Out of the 94 subjects, 53 are male, 40 are female, and 1 is non-binary, and their ages range from 39 to 96 years, with an average age of 66.7 years and a standard deviation of 14.4 years. Most of the images were acquired at AZ Groeninge in Kortrijk, Belgium using a GE medical system, although the dataset is not restricted to this hospital.

The high-resolution CT images allow for detailed segmentation. The slice thickness and spacing of the images vary. The slice thickness, which reflects the resolution in the z-direction, is either 0.625mm or 1.250mm. The spacing, which reflects the resolution in the x- and y-direction, varies in the range between 0.578mm and 0.971mm. The image size is not always equal but each image has around 2000 slices, often of a size of (512,512). For a more detailed overview of this information, please refer to A.2.

Alongside the CT images, 3D ground truth models of the bones in the lower limb region are also available. Models of the pelvis, femur, tibia, and fibula are provided, with separate files provided for the left and right sides, resulting in a total of six files. The pelvis is given as one model but is preprocessed to split the left and right hip bones. Note that one hip bone is also referred to as the pelvis in this thesis to simplify things. Each model is provided in the .obj file format, which is widely used for 3D models. The file contains a mesh that describes the object in terms of vertices and faces. Each vertex is represented by a 3D coordinate, and a face is defined as a combination of 3 connected vertices. The meshes of the same bones, regardless of the left or right side, correspond with each other. Correspondence is found by applying the method of Audenaert et al. [1]. The 3D models provide information on both the inner and outer layers of the cortical bone. In this this, only the outer cortical layer is used. Table 3.1 provides the edge length and the number of vertices and faces of the outer layer. By taking the image origin and spacing into account, the location of the 3D models does match the correct location in the images.

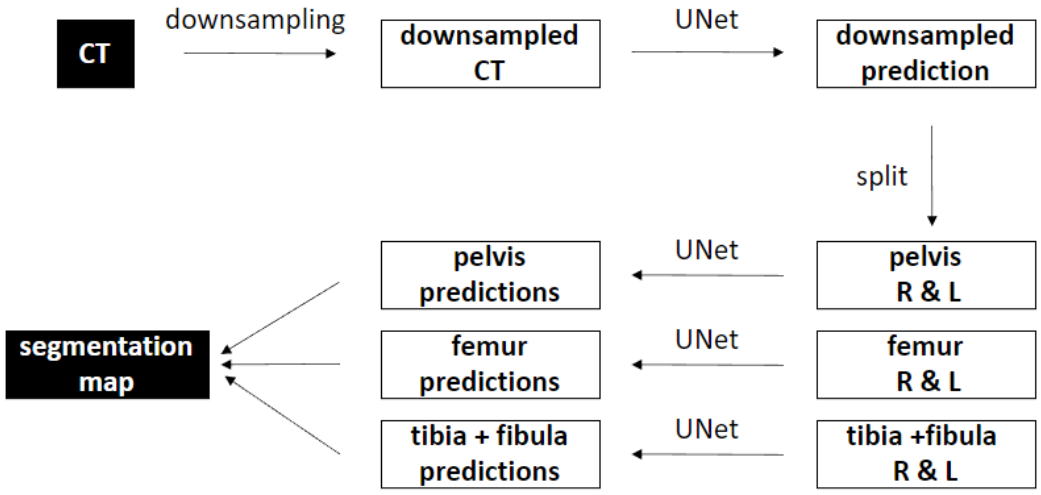


Figure 3.1: Proposed segmentation workflow.

3.2 Full segmentation worfklow overview

This section describes the workflow proposed to generate precise subject-specific 3D models of the lower limbs from a full CT image. First a general overview of the process is provided, followed by a detailed explanation of each step, given in the following sections.

A more general overview is shown in Fig. 3.1. The workflow takes a full CT scan of a subject and segments the bones of the femur, pelvis, and tibia combined with the fibula, further visualised in Fig. 3.2. The goal is to obtain a segmentation map of the bones at full resolution, where each voxel in the input CT image is labelled as background or a bone. These maps can be converted to 3D models. The bones are segmented separately for the left and right sides, resulting in a total of six bone classes in the final segmentation map. To accomplish this goal, we propose utilising the UNet model. However, due to its high computational demands, the UNet cannot handle full CT scans in a single operation. Hence, it is necessary to devise a method for dealing with smaller images.

One approach would be to do patch-based segmentation on the input image, dividing the image into smaller patches or sub-images and performing segmentation on each patch separately. However, this technique may lead to a possible loss of contextual information. As each patch is processed separately, no information on neighbouring patches is included leading to possible discontinuities or artifacts, especially at the patch boundaries. Next, the inference of a patch-based prediction can take a long time due to the need for processing multiple patches separately and then reconstructing the final output. To address this issue, the objective of the workflow is to enable the processing of complete images as a whole, eliminating the requirement for

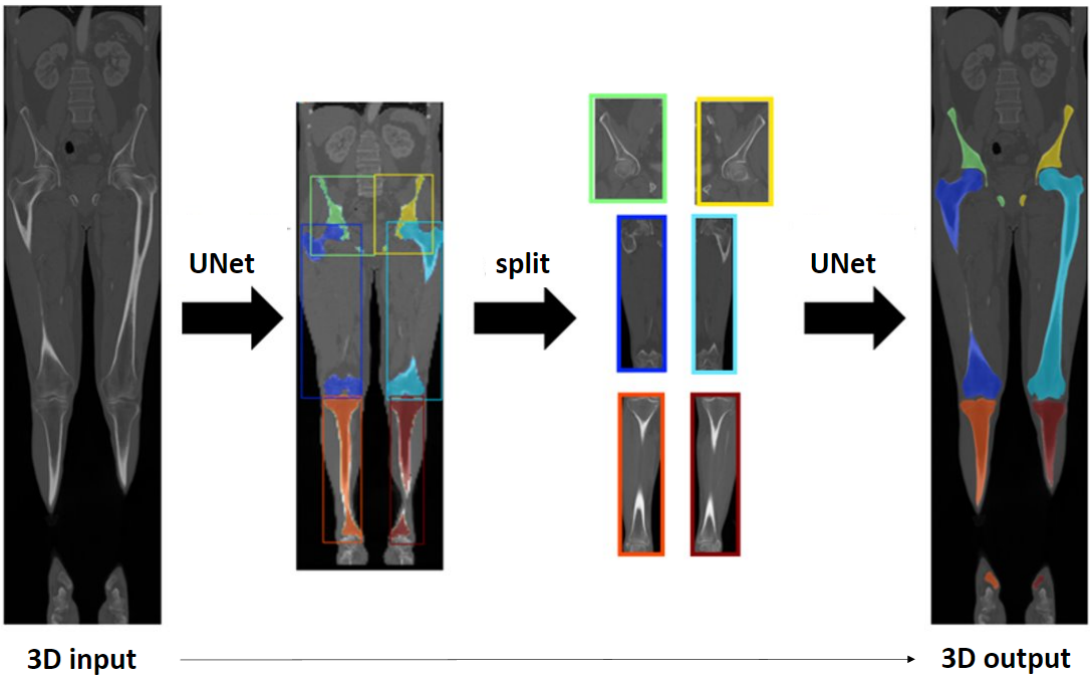


Figure 3.2: Segmentation workflow visualised.

patch-based segmentation. To accomplish this, an alternative approach is required for handling sub-images. First, the general locations of each bone of interest in the CT image are located, allowing cropping the full-resolution CT image to the ROI for each bone. This is done by using a downsampled version of the original image. A UNet network is trained on these downsampled images to obtain a multiclass segmentation of the bone of interest. After completing the multiclass prediction on the downsampled image, it becomes possible to locate the bounding box of the bones in the full-resolution CT. This enables the cropping of the image into six smaller sub-images, with each sub-image focusing on one bone of interest.

Finally, independent UNet networks for the femur, pelvis, and tibia combined with the fibula are trained and applied to the sub-images to obtain the final predictions, which can be merged back into the original CT image. This approach allows processing full images of the ROI, resulting in precise subject-specific segmentation maps of the bones in the lower limbs. Note that while the end goal is to obtain 3D bone models, this workflow focuses on generating segmentation maps as an intermediate step. Although segmentation maps can be converted to 3D models, it is not the main focus of this workflow in this chapter. The predicted label maps are later converted to 3D models, discussed in Chapter 4.

Label	Type
0	background
1	femur R
2	femur L
3	pelvis R
4	pelvis L
5	tibia + fibula R
6	tibia + fibula L

Table 3.2: Labels and their corresponding structure.

3.3 Bone localisation

The first step of the workflow is to localise the femur, pelvis, and tibia combined with the fibula in downsampled CT images and cropping the original CT image to the ROIs, based on this localisation. This section explains how this is achieved, starting from the original data. This gives a complete overview of how the technique to obtain bone localisation is developed and can be applied to unseen input images.

3.3.1 Data preprocessing

Label Maps

The proposed method uses the UNet network to obtain segmentation maps, a matrix of voxels that assigns a class label to each voxel in the image, indicating which part of the image belongs to which anatomical structure. In order to train a UNet, the label maps corresponding to the CT images are needed. These label maps serve as ground truth data, providing the correct segmentation information for each voxel in the input image. Since the available data comprises 3D models of the bones, a method of converting these models into label maps is required. It is important that the created maps are of good quality, meaning that the voxels should be labelled correctly so that the network can learn to segment accurately.

For each subject a label map is created, the same size as the original CT image. Each voxel is labelled with a value from 0 to 6, depending on the underlying structure. The label values and their corresponding bone type can be found in Table 3.2. A label map is generated by using both the CT image and the 3D models as input. The script then produces the desired label map as the output.

Before creating the label maps, the slice thickness and spacing of the images, making up the

voxel size, are all set to 1mm. This is done by applying the linear interpolation method from ShapeWorks to change the voxel size of the CT images [47]. It should be noted that this interpolation may lead to a loss of details when the original slice thickness or spacing is smaller than the new value of 1mm. Nevertheless, bringing the CT scans to a consistent voxel size can help standardise the data and facilitate more stable segmentations when using a CNN. This is because the convolutional kernels, which have a fixed size, always get the same physical information when sliding over images with equal spacing. Hence, with the interpolation, a tradeoff is made between more standardised data and the possible loss of details.

Once the images have an equal voxel size of 1mm in each direction, the mesh voxelization script from Adam A. is applied [49]. This gives the final segmentation maps in .mat file format. As three of the image could not be converted to label maps, due to memory issues, 92 subjects are left. The patient IDs that are not used are N76, N89, and N92.

Downsampling of the images

Downsampling of the images happens in a very straightforward way. An image is downsampled by sampling each 5th voxel in the z-direction (slices) and every 4th voxel in the x and y-direction. The created label masks are downsampled in the same way.

Downsampling the image based on an interpolation technique was also considered, but this does not seem to improve results. Hence, this technique is not further applied.

Image padding and normalisation

An extra preprocessing step is performed by applying padding to each CT image. This padding involves adding extra voxels around the image with a value of -1024.0. The reason for padding is to ensure that the image size is divisible by 8 in each direction. This is necessary because the UNet is designed to perform both down- and upsampling operations. Each of the downsampling operations reduces the size of the image by a factor of 2 in each direction. To ensure that the final output after upsampling remains the same size as the input, the input size should be divisible by 2 to the power of the number of downsampling operations used. In this case, the proposed network uses 3 downsampling operations, resulting in a reduction factor of 8 in each direction. Therefore, the original image size should be divisible by a factor of 8 in each direction to ensure that the final output after upsampling remains the same size as the input. The value of -1024 is chosen because it is equal to the background information. Hence, padding this value should not influence the information in the images itself. Each label mask is padded in the same way, but with a value 0, depicting the background.

The final preprocessing step before feeding the image to the UNet is normalisation. Normalisation is a crucial preprocessing step in training deep learning models, especially for image classification tasks using CNNs. It helps to standardise the dataset, making it easier to learn

and generalise the patterns in the data. Normalisation involves converting the data to a common range or scale, bringing consistency to the training procedure. In this work, each image was normalised by applying the following equation:

$$image(x, y, z) = \frac{image(x, y, z) - mean(image)}{std(image)}. \tag{3.1}$$

Each value is divided by the standard deviation of the image after subtracting the mean of the image, a common technique called zero-mean normalisation. This technique standardises the data by centring it around zero and scales it based on the standard deviation.

3.3.2 Designed UNet architecture

Once images are preprocessed and label masks are ready, both can be fed to the UNet. This part explains how the UNet architecture, implemented in PyTorch, looks in more detail [50]. This architecture is used for the bone localisation and the bone segmentation. The proposed UNet is based on the one proposed by Ambellan et al., which uses their architecture for the prediction of the femur and tibia in the joint region using the Osteoarthritis Initiative (OAI) dataset, working on MRI data [8]. Some minor changes to their network are applied. An overview can be seen in Fig. 3.3.

The implemented architecture includes an encoder and a decoder with a bottleneck block in between. The encoder is made up of three convolutional blocks, each containing two 3D convolutional layers with a kernel size of 5 and padding of 2. After these layers, instance normalisation and a ReLU activation function are applied. The bottleneck block consists of a single convolutional block, with the same structure as the encoder convolutional block. The number of filters in the first convolutional layer of the first block is 16, and this number doubles in each subsequent layer until the bottleneck, where there are 128 filters. The encoder also features a 3D max pooling layer with a kernel size of 2 to downsample the images.

The decoder is made up of three upsampling blocks, each containing a 3D transposed convolutional layer with a kernel size of 2 and stride of 2, followed by concatenation with the corresponding encoder block feature map. The concatenated feature map is then passed through a convolutional block similar to the one used in the encoder but with the number of filters reversed. Finally, the output of the last convolutional block in the decoder is passed through a 3D convolutional layer with a kernel size of 1 and padding of 0 to produce the segmentation map. The number of classes can be specified, indicating how many output layers are produced. In the case of bone localisation using multiclass segmentation, 7 output classes are used.

The proposed network differs from the one in Ambellan et al. in two ways. First, the proposed network uses fewer convolutional filters, ranging from 16 to 128, while in Ambellan et al. it

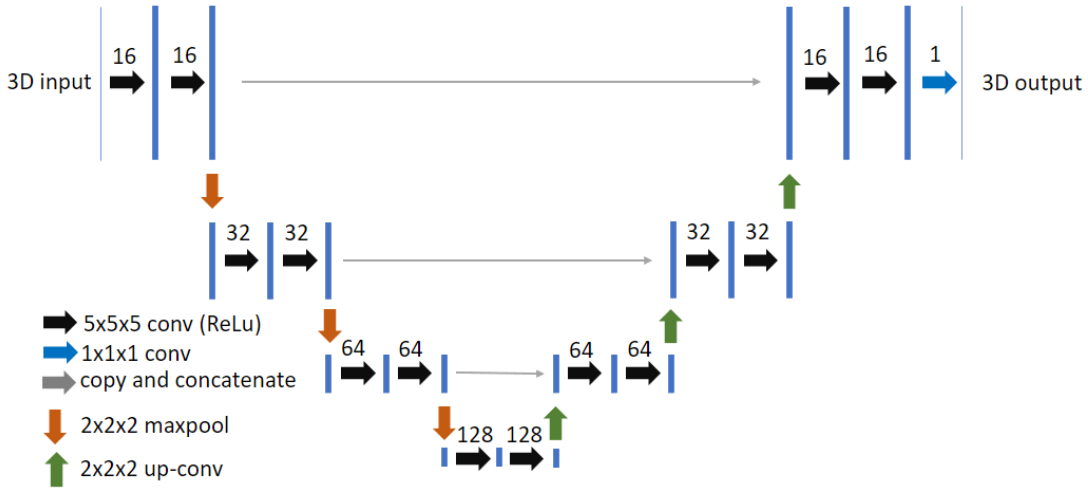


Figure 3.3: Designed UNet architecture.

ranges from 32 to 256. During training, no visible improvement is observed with a larger number of filter kernels, so a lower amount is used to reduce memory usage, allowing working with larger-sized images. Next, no dropout is used during training of the proposed networks. The network of Ambellan et al. does use a dropout approach, which randomly ignores certain parameters during training, to help prevent overfitting. The effect of leaving out the dropout technique is not investigated.

3.3.3 Training method

Data split

In order to effectively train, validate, and test the network, it is crucial to properly split the dataset. The subjects whose CT image and label maps are available are divided into three distinct groups. The first group, consisting of 72 subjects, is dedicated to training the model and enabling it to adjust its parameters. The second and third groups, containing 10 and 9 subjects, are respectively assigned to validation and testing. The validation set plays a key role in tuning the hyperparameters and monitor the behaviour of the model to prevent overfitting during training. The test set is used to evaluate the model its performance on previously unseen cases, given in Chapter 4. Specifically, the validation set includes subjects N1 to N11, and the test set includes subjects N12 to N21, with the remaining subjects reserved for training purposes. A fixed split is opted for as it provided a clear and convenient way to divide the dataset, which could be consistently used throughout this thesis. Additionally, this split is also used in the augmentation technique which will be discussed later.

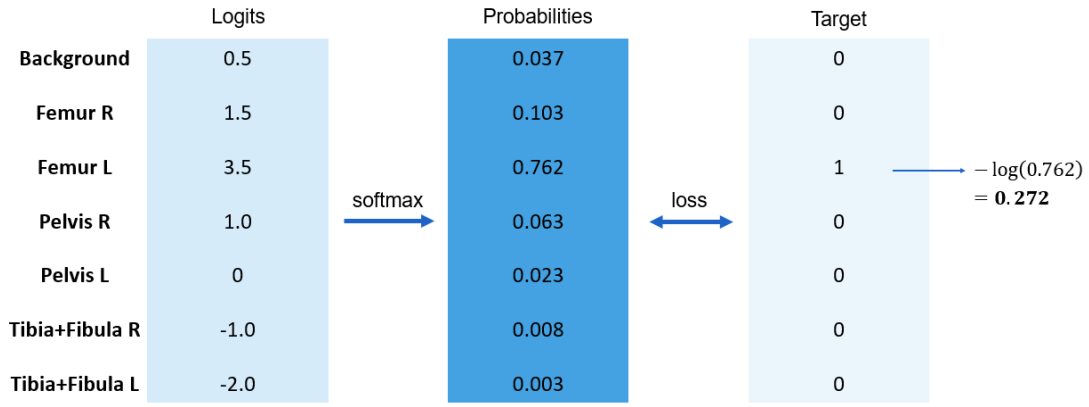


Figure 3.4: CE loss calculation for one voxel.

Loss function

During training the loss function is optimised by comparing the predictions with the ground truths or targets, being the label masks. This is done by using the cross entropy (CE) loss, a common loss used in multiclass classification tasks. The cross entropy is defined as the negative log-likelihood of the predicted probability distribution over all the classes, given the true probability distribution.

The PyTorch CrossEntropy function is used, which combines the softmax activation function with the CE loss in one, making it easy to use. This means that the function automatically compares the logits, which are the output values from each layer, to the target values by first applying the softmax activation function, giving a probability value per class, and then evaluating the cross entropy. This is applied to each voxel. The CE for a voxel at position (x,y) when having n output classes is defined as

$$L_{CE,(x,y)} = -\sum_{i=1}^{n} y_i(x,y)\log(p_i(x,y)), \tag{3.2}$$

where y_i is the true label of the i-th class (0 or 1), p_i is the predicted probability of the i-th class, and the sum is taken over all n classes of the voxel output. To get a single loss value for an image, the mean value of all voxels is taken. For an image with size (m, n), the following equation can be applied:

$$L_{CE,image} = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} L_{CE,(x,y)}}{m \cdot n}. \tag{3.3}$$

An example of the loss calculation for a single voxel is done can be seen in Fig. 3.4. The logits are the output of the UNet for the voxel. After applying the softmax activation and Eq. (3.2), a CE loss of 0.272 is found. One can observe that a higher probability for the correct class leads to a lower loss value. Hence, this is a good measure to optimise the network.

As mentioned, the CE loss for the complete image involves computing the mean of the losses of all voxels in the image, seen in Eq. (3.3). However, this approach needs to be modified because there is a class imbalance in the label masks. In such cases, where most voxels belong to the background class (class 0), the background loss can dominate the overall image loss, leading to the model being biased towards the majority class. To address this issue, the weighted cross entropy (WCE) loss is used, which assigns a weight to each class. For an image of size (m, n) this can be defined as

$$L_{WCE,image} = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} L_{CE,(x,y)} w_{gt(x,y)}}{m \cdot n}, \tag{3.4}$$

where $w_{gt(x,y)}$ is the weight for the ground truth class of pixel (x, y). The weight w of each class determines its contribution to the overall loss function, such that the model is forced to give more attention to the minority class during training. The weights of all classes are set to 1 and that of the background to 1/10. By using WCE, the performance of the model on the minority class is improved, without compromising its ability to classify the majority class.

Optimiser and hyperparameters

The training of a UNet comes automatically with a large number of hyperparameters that need to be tuned. The hyperparameters of the UNet architecture are already mentioned in the previous part. These are the kernel size, padding, stride, number of layers, and convolutional filters. These are all fixed to their mentioned values and not much further tuning is done, as this did not seem to significantly change the found results. Next to these, the optimiser and other hyperparameters need to be set for training the network. The Adam optimiser is used in this thesis. The other hyperparameters are the number of epochs, batch size, and learning rate. Because the images are quite large, the only possible batch size is 1. Even though a larger batch size may benefit the generalisation, this did not seem to be an issue. The learning rate and epochs still need to be set. More information on the final values can be found in Chapter 4.

3.3.4 Predictions and bone localisation

The trained UNet model can be used to generate segmentation maps for new, unseen CT images. The goal of segmentation is to identify and localise the femur, pelvis, tibia, and fibula in the image. To apply the model to an unseen image, it is preprocessed in the same way as the training images. This involves downsampling, normalisation, and padding to ensure that the input size and format match the model its requirements.

The preprocessed images are passed through the UNet model to generate a prediction. Since the model has seven output channels, the output is converted into a segmentation map. This is achieved this by selecting the class corresponding to the layer with the highest logit value for

each voxel. This generates a segmentation map for the downsampled image, which is used to localise the bones in the original image.

To localise the bones, the coordinates of the downsampled image are mapped to the original image. This is done by taking into account the size reduction and padding applied during preprocessing. Using this information, the bounding box for each bone can be identified in the downsampled image. Then the bounding box coordinates in the downsampled image are converted to the original image coordinates, completing the bone localisation process. By following these steps, the UNet model can be used to generate accurate segmentation maps and localise bones in new CT images.

3.4 Bone segmentation

After localisation is completed, the goal is to obtain accurate segmentations of the bones in the original resolution. Note that the original resolution here is set to 1mm, as the label maps are created in this way. To do this, three separate networks are trained. One for the femur, one for the pelvis, and one for the tibia combined with the fibula. To complete this, three separate training processes are done. In this section, the needed data preprocessing, the training process, and the method for obtaining the final segmentation map are explained.

3.4.1 Data preprocessing

After the bones are localised, the following step is to crop the full CT images. Six sub-images are created from the full CT image, one for each bone class. Cropping happens based on the found bounding boxes, but the bounding boxes are enlarged with 10 voxels on each side to make sure that the wanted bone is completely located within the sub-image, even when a small prediction error is made on the edges of the bones in the downsampled image. The label masks are cropped in the same way.

While there are six bone classes, three models are created, making use of the similarity between the right and left sides. The models are trained on the right side. The left side is mirrored to create extra training samples, giving 144 training samples instead of the original 72.

After cropping and mirroring, the same normalisation and padding procedure as explained before is applied.

3.4.2 Training

The training procedures for the three models are very similar to that of the bone localisation. The same UNet architecture and hyperparameters are used. Hence, these will not be repeated in this section. The only difference in the UNet architecture is the number of output classes. To predict only one class at a time instead of multiple classes, the number of output classes is set to 1. Due to the change in the number of classes, a different loss is used too. Now the binary cross entropy (BCE) loss is used, implemented in PyTorch as BCELoss. This is the same as the CE loss but applied to binary classification. Furthermore, the number of epochs also differs depending on the trained model.

3.4.3 Final predictions

After training the three models, the workflow can be completed. Firstly, the bone localisation is performed followed by the cropping of images using the previously explained method. Then, these sub-images are passed through the corresponding trained bone models using a forward pass. The output of each forward pass is then converted into a segmentation map, where every voxel with a prediction higher than 0.5 is set to 1, depicting the bone structure, and the rest is set to 0, representing the background. As the models are trained on the right side, it is necessary to mirror the bones of the left side before the forward pass. Finally, by combining all the predictions and place them on the original CT image, the workflow is completed. In Chapter 4, further evaluation of this workflow is done.

3.5 Evaluation metrics

An important task is quantifying the results using the appropriate evaluation metrics. This is an essential step to see how well the workflow performs. This section explains the used metrics in further detail. These metrics are used to evaluate both the bone localisations and bone segmentations. For the bone segmentation, it is important to notice that the metrics are evaluated on the full CT image and not on the cropped versions. The metrics will be defined for each bone separately. Hence, for each voxel, it can be stated if it has been classified correctly or not. A voxel classified as the bone of interest is stated to be positive, otherwise, the prediction is negative. This leads to 4 possible classes when comparing it to the ground truth. These are true positive (TP), true negative (TN), false positive (FP) and false negative (FN). An overview can be seen in Fig. 3.5. These classes are important to define the metrics.

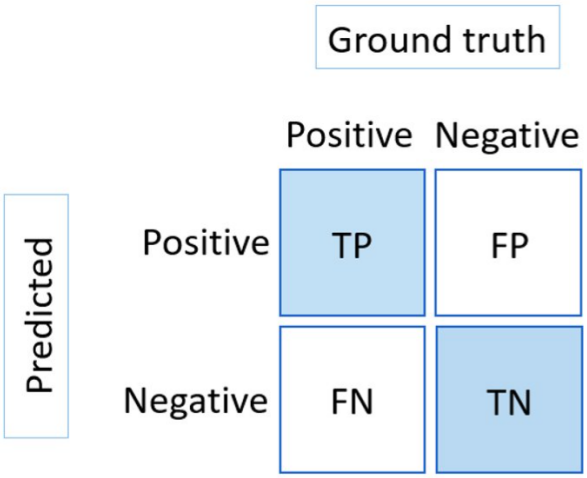


Figure 3.5: Possible classification classes per voxel.

3.5.1 Dice similarity coefficient

The Dice similarity coefficient (DSC), also known as the Sørensen–Dice coefficient, has become a very popular metric to evaluate semantic segmentation results [51, 52]. One of the first to propose the DSC for medical image segmentation were Zijdenbos et al., showing that it is a good metric when working with a class imbalance and noting that it both has good localisation and size agreement with perceptual evaluation [53].

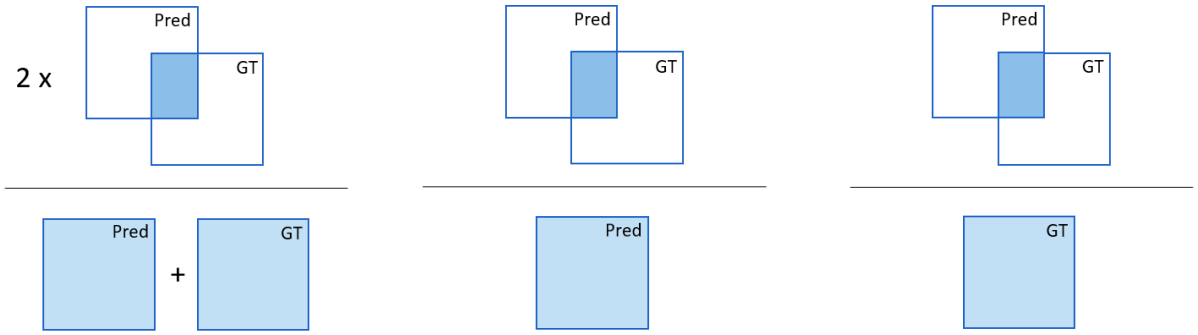
The DSC is defined as twice the size of the intersection of the two sets, being the prediction and ground truth here, divided by the sum of the sizes of the two sets:

$$\mathrm{DSC} = \frac{2\,|Pred \cap GT|}{|Pred| + |GT|} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{3.5}$$

$Pred$ is the prediction set and GT the ground truth set. $|\cdot|$ depicts the cardinality, i.e. the number of elements in the set. The description in terms of the explained classification classes is also given. The DSC has a value between 0 and 1, with 1 being a perfect overlap of the prediction with the ground truth. As the metric does not rely on TN, a class imbalance where the negative class is dominant, does not bias the performance. For the used medical data, the majority of the image data represents the background and only a small fraction of the voxels are the bones to be segmented. This indicates the presence of imbalanced data. Therefore, it is important to use appropriate metrics, such as the DSC, that account for the data imbalance.

3.5.2 Precision and recall

Two other metrics that are defined to deal with imbalanced data and are based on the classification classes are precision and recall. Both give information about different aspects of the



(a) Dice similarity coefficient (b) Precision (c) Recall

Figure 3.6: Visualisation of the used evaluation metrics.

model its performance. The DSC both evaluates both under- and oversegmentation, these metrics evaluate it separately. Both scores are measured between 0 and 1, with 1 being a perfect score. Precision is a measurement of the accuracy of the positive predictions and is defined as the fraction of the number of voxels that are correctly classified as positive and the total amount of voxels that are classified as positive. This can be written as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3.6}$$

A high precision score indicates that the network produces few false positives. It is important that the network does not incorrectly classifies voxels as bone structures when this is not the case. Otherwise, this could for example lead to an incorrect diagnosis or wrong treatment planning. Hence, precision is a useful measurement.

Next, recall, also known as sensitivity, is a measurement of the completeness of the prediction. Recall is defined as the fraction of the voxels that are correctly classified as positive and the total amount of voxels that are positive in the ground truth, given as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3.7}$$

A high recall score indicates that the network can identify most of the voxels belonging to the target structure. Hence, a high recall score is needed to indicate that there is no underestimation of the bone structure, which could also lead to false clinical conclusions.

A visualisation of the DSC, precision and recall is shown in Fig. 3.6.

3.5.3 Hausdorff distance

The previous metrics make use of the overlap between the prediction and the ground truth, but segmentation is about more than that. A good model is able to capture the correct shape and boundaries of objects. Therefore, the last evaluation is the Hausdorff distance (HD) [54]. Given two sets A and B, the HD is defined by

$$\mathrm{HD}(A, B) = \max(h(A, B), h(B, A))), \tag{3.8}$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||. \tag{3.9}$$

$|| \cdot ||$ is an underlying norm of A and B and $h(A, B)$ is called the directed Hausdorff distance from A to B. In this work, A and B can be replaced with GT and $Pred$, and the used norm is the Euclidean norm. This metric is given in millimetres, knowing the resolution of the images.

This shows that the Hausdorff distance is an indicator of the biggest segmentation error. It can be seen as the furthest distance between a point in one of the two sets to its closest point in the other one. This gives a better understanding of the possible shape error of the predictions. A large HD can also indicate that certain voxels in the image, far from the ROI, are misclassified.

3.6 Conclusion

This chapter outlines the proposed method for generating subject-specific segmentations of the pelvis, femur, and tibia combined with the fibula, presented as a segmentation map. These maps can then be converted into 3D models in the subsequent stage. The proposed method consists of two main components: bone localisation and bone segmentation. In order to create a useful tool, it is crucial that a full CT scan is provided as input. However, processing a complete CT scan at once can be challenging due to its size. To address this, a bone localisation method is proposed that analyses a downsampled image to determine the general location of the bones. Subsequently, sub-images are created to perform more precise segmentations. Combining these steps enables producing an accurate output segmentation map by processing the full CT scan given as input.

4 Lower limb bone segmentation results and discussion

This chapter presents the results of the lower limb bone segmentation workflow, explained in Chapter 3. The workflow is applied on the angio-CT data, making use of the proposed UNet architecture for both the bone localisation and bone segmentation. Section 4.1 shortly discussed the training procedure. Results of both the bone localisation and bone segmentations are given in Section 4.2. Finally a discussion of these results is given in Section 4.4.

4.1 Training process

Four models are trained in total, 1 on the downsampled image and one for each kind of bone. Even though each of the trained models has the same UNet architecture, other hyperparameters to achieve the best possible result for each model need to be set. All of the models are trained on a server available at the University Hospital of Ghent (UZ Gent), using a Tesla V100-PCIE-16GB GPU. The batch size is fixed to 1 and the learning rate is set to 0.001 for each training procedure. Furthermore, only the number of epochs differs depending on the model. An overview of the validation loss for each training, also showing the final number of epochs, can be found in Fig. 4.1. Each epoch takes around 1200 seconds. It can be seen that the validation loss drops nicely for each training. The training of the femur took the longest, having 60 epochs, but also reaches the lowest loss value of 0.00477. The other final validation loss values in rising order are 0.00724, 0.00760, and 0.0107, with the highest value for the tibia combined with the fibula model.

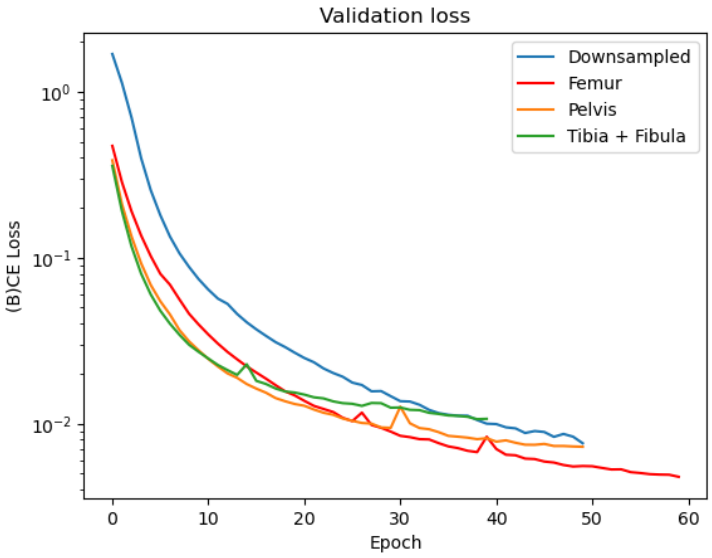


Figure 4.1: Validation loss of each trained model.

4.2 Results: bone localisation

In this section, the results of the bone localisation step on downsampled images are presented, using a combination of evaluation metrics from Section 3.5 and visualisations of the predictions. The evaluation metrics are summarised in Table 4.1, showing mean and standard deviation values for each class. The analysis reveals that the DSC for femur and tibia combined with fibula is high, ranging from 0.934 to 0.952, with low standard deviation. However, for the pelvis, the DSC values are lower, with mean values of only 0.88 and 0.891.

Although recall values are consistently high for every class, ranging from 0.987 for the left pelvis to above 0.99 for other classes, the precision values are clearly different, ranging from 0.790 to 0.906. In particular, the precision of the pelvis is very low, with mean values of only 0.790 and 0.813. When the recall is high but precision is low, it indicates that the classifier is correctly identifying a high proportion of positive cases but also misidentifying a large number of negative cases as positive, resulting in a large number of FP. Therefore, the network trained on the downsampled images appears to be too sensitive, especially for the pelvis.

Further clarification is needed to understand the found HD values. The average HDs for the femur and right tibia combined with the fibula, fall within a range of 8.6mm and 12.4mm, with relatively small standard deviations. The HD represents the maximum distance between the predicted location and the actual location. The objective is to create a bounding box around the ROIs, and an error of up to 10mm is acceptable since it does not significantly impact the appearance of the sub-images. It is important to note that the bounding boxes are padded with

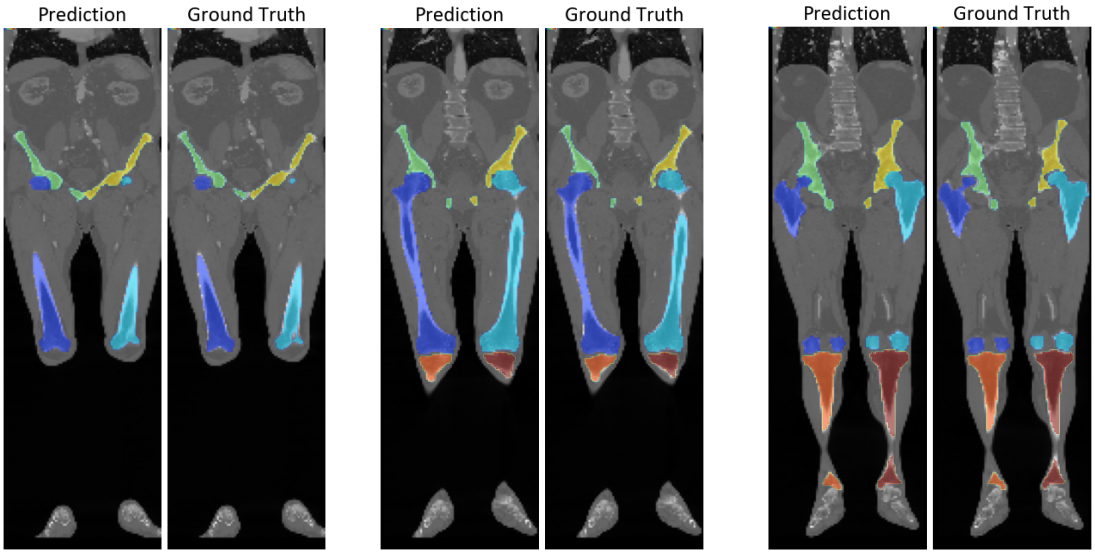
Type	**DSC**	**Precision**	**Recall**	**HD (mm)** (without outliers)
Femur R	0.949 ± 0.011	0.906 ± 0.020	0.995± 0.003	10.2 ± 4.48 (9.11 ± 3.21)
Femur L	0.952 ± 0.008	0.915 ± 0.015	0.994 ± 0.004	8.60 ± 1.92 (8.04 ± 1.20)
Pelvis R	0.880 ± 0.030	0.790 ± 0.047	0.996 ± 0.003	31.7 ± 48.5 (14.7 ± 7.04)
Pelvis L	0.891 ± 0.025	0.813 ± 0.030	0.987 ± 0.021	30.0 ± 85.7 (12.7 ± 3.69)
Tibia + Fibula R	0.934 ± 0.016	0.880 ± 0.027	0.995 ± 0.004	12.4 ± 2.32 (12.0 ± 2.22)
Tibia + Fibula L	0.936 ± 0.014	0.884 ± 0.023	0.994 ± 0.004	218 ± 219 (9.29 ± 1.56)

Table 4.1: Metrics of bone localisation predictions.

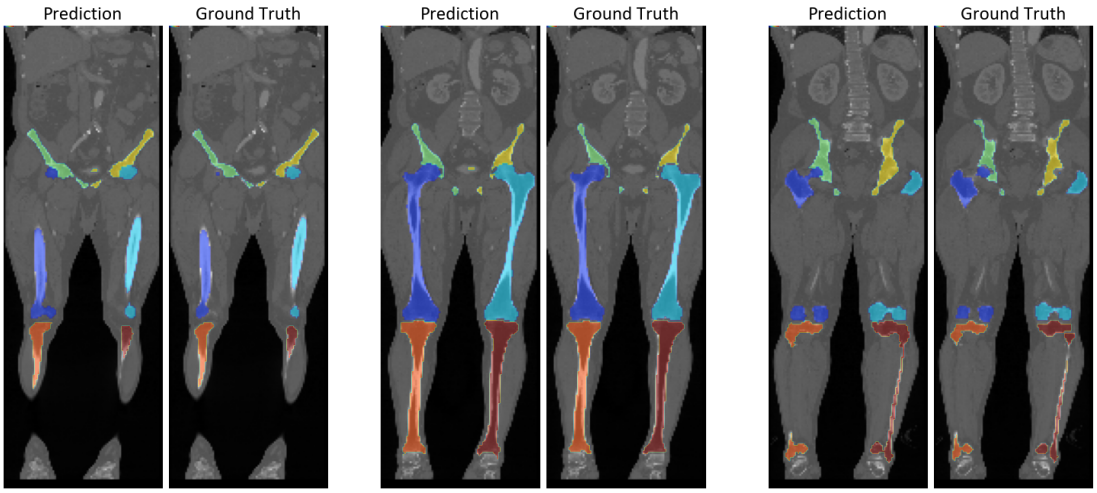
an additional 10mm on each side to ensure the ROI remains within the sub-image, even if the errors are underestimated. However, based on other measurements, it seems that the network is too sensitive, leading to believe that the errors are likely caused by overestimation. The mean HDs of the other classes exhibit high values, accompanied by large standard deviations. These results are due to two particular cases, N18 and N15. In these instances, a few voxels that are not in proximity to the ROIs are incorrectly predicted, resulting in a high HD. This anomaly is responsible for the high values observed in these cases. For N18, this occurrence takes place in all classes with a high mean HD, whereas for N15, this happens only with the left tibia combined with the fibula. These few voxels will also impact the bounding boxes, which will be explained later. To demonstrate that these large mean and standard deviation HD values are due to these two cases, the means and standard deviations while excluding N18 and the tibia combined with the fibula value of N15 are included too.

Fig. 4.2 depicts the predictions and ground truths of a selection of test cases, showing 3 coronal slices for each test case. While the metrics for most cases are very similar, varying around the mean values presented in Table 4.1, 2 random subjects for visualisation are chosen: N12 and N21. Notably, one case, N18, shows significant deviation from the other cases in terms of DSC, precision, recall, and HD for all classes, particularly for the pelvis, where precision values of 0.762 and 0.797 were obtained for the right and left sides, respectively. This indicates a large number of false positives for the pelvis in this case. The DSC and Recall values are also lower for all classes. N18 is also the case that causes the high mean values and standard deviations of the HD for some of the classes by the few voxels that are wrongly predicted, as explained above. The visualisations confirm the statements made earlier. In general, the predictions are highly accurate, with minimal FN, resulting in high recall values. Additionally, the lower precision values for the pelvis are evident. While the general shape of the pelvis can be located quite well, there is a tendency to overestimate when comparing predictions to the ground truth, resulting in lower precision. Nonetheless, all predictions, including those for the pelvis, are highly accurate across all instances. The high HDs are more difficult to visualise as one single voxel predicted wrongly, which is difficult to see, can lead to these high values.

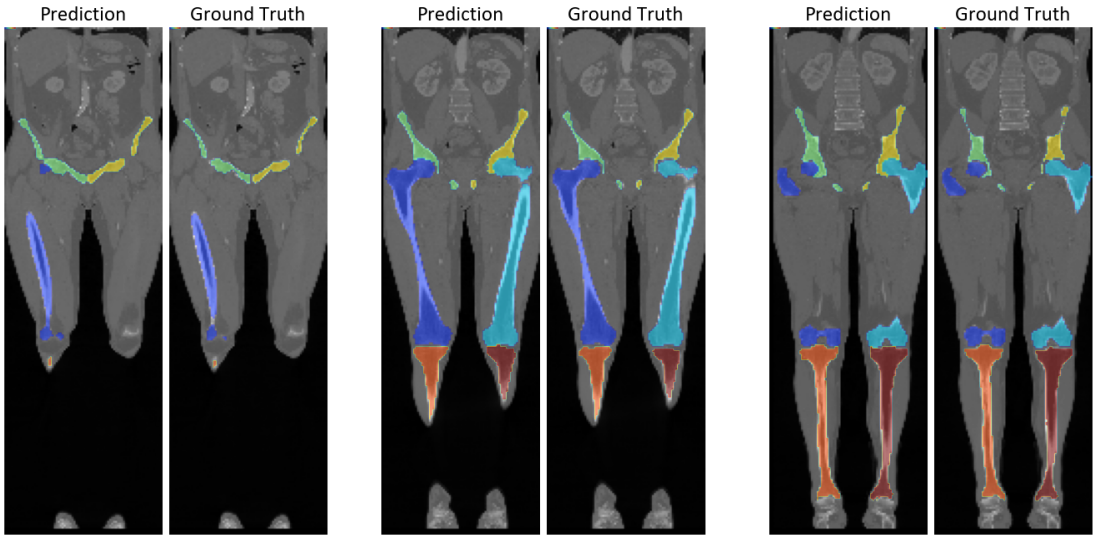
From these predictions, the bounding boxes are calculated. Once bounding boxes are found, the



(a) Slice 40 (N12) (b) Slice 45 (N12) (c) Slice 50 (N12)

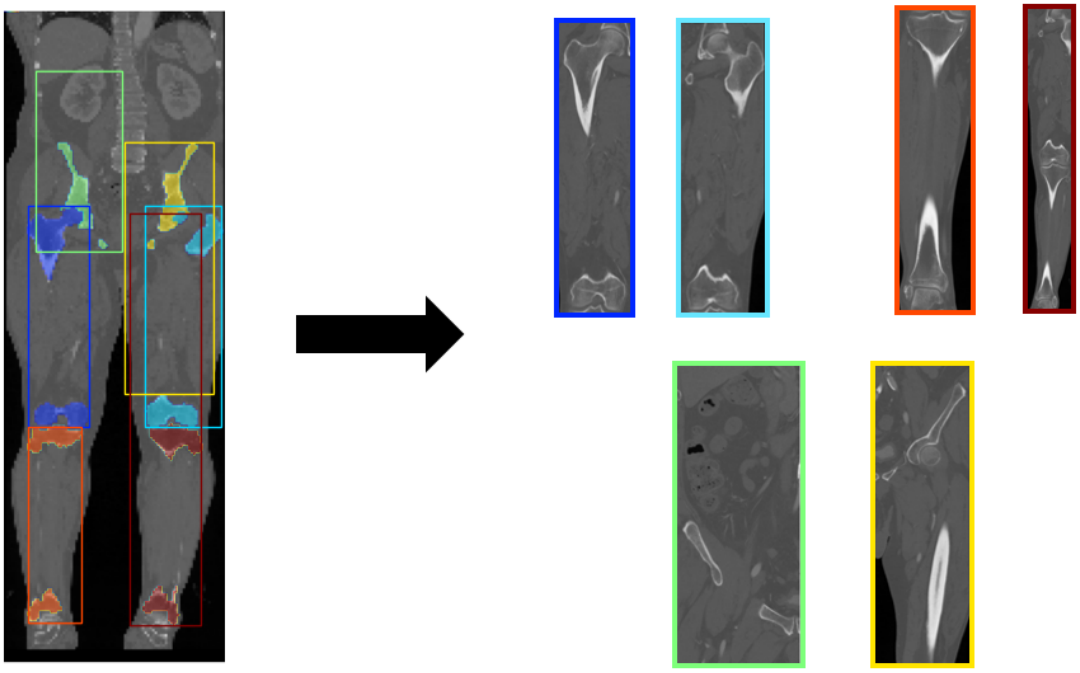


(d) Slice 40 (N18) (e) Slice 45 (N18) (f) Slice 50 (N18)

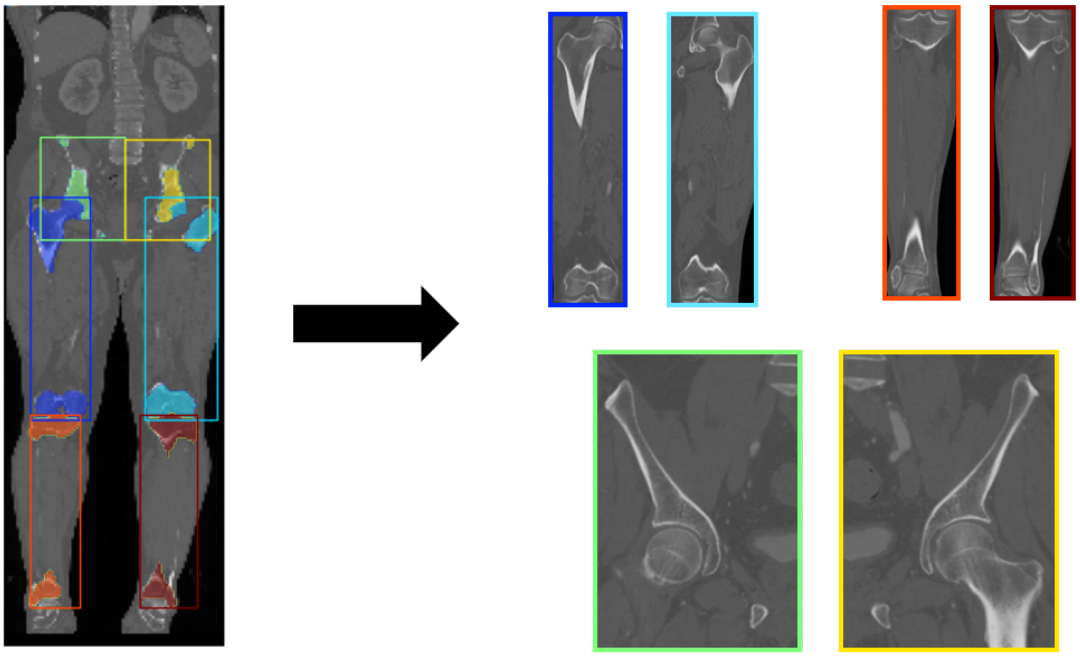


(g) Slice 40 (N21) (h) Slice 45 (N21) (i) Slice 50 (N21)

Figure 4.2: Results bone localisation for N12, N18 and N21. Slices given in coronal direction.

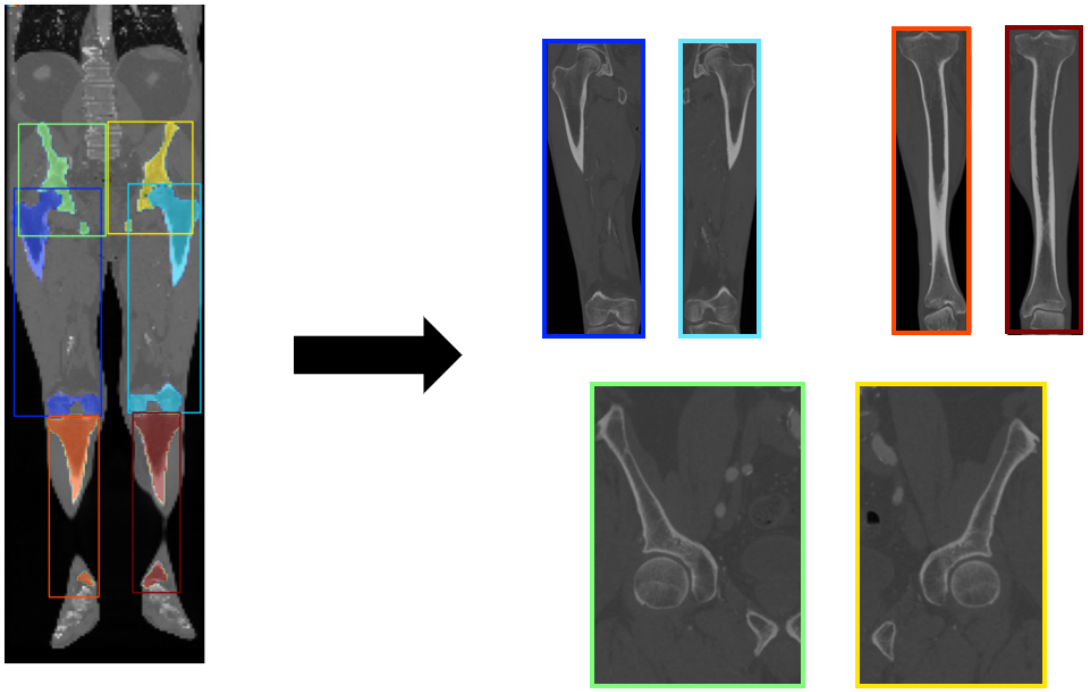


(a) N18 before median filtering

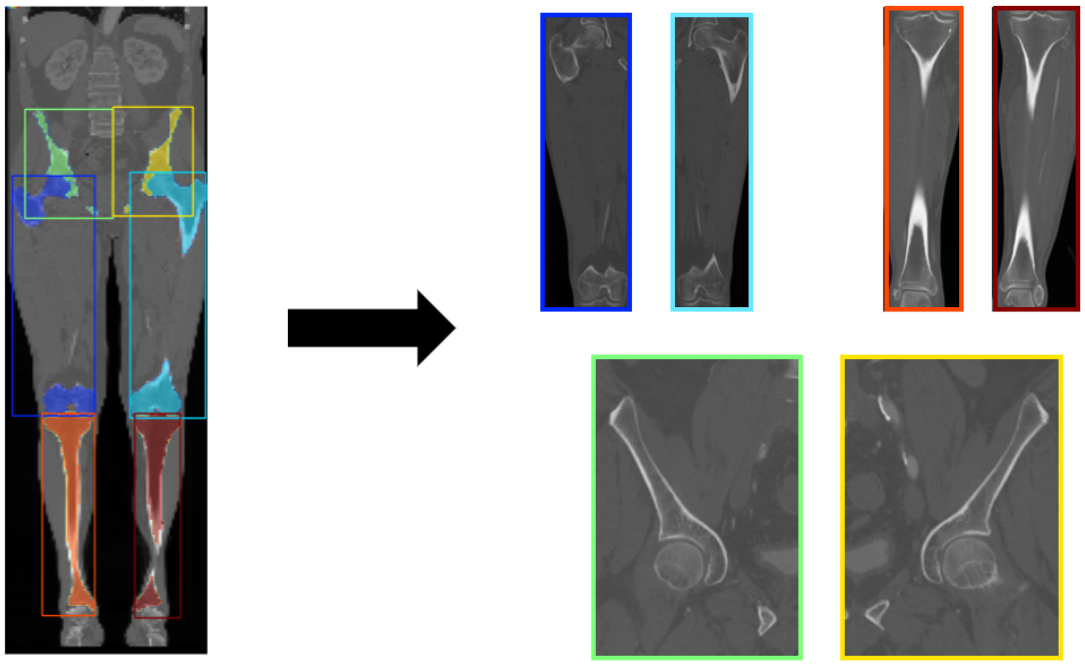


(b) N18 after median filtering

Figure 4.3: Bounding boxes and resulting sub-images before and after median filtering: N18.



(a) N12 after median filtering



(b) N21 after median filtering

Figure 4.4: Bounding boxes and resulting sub-images after median filtering: N12 and N21.

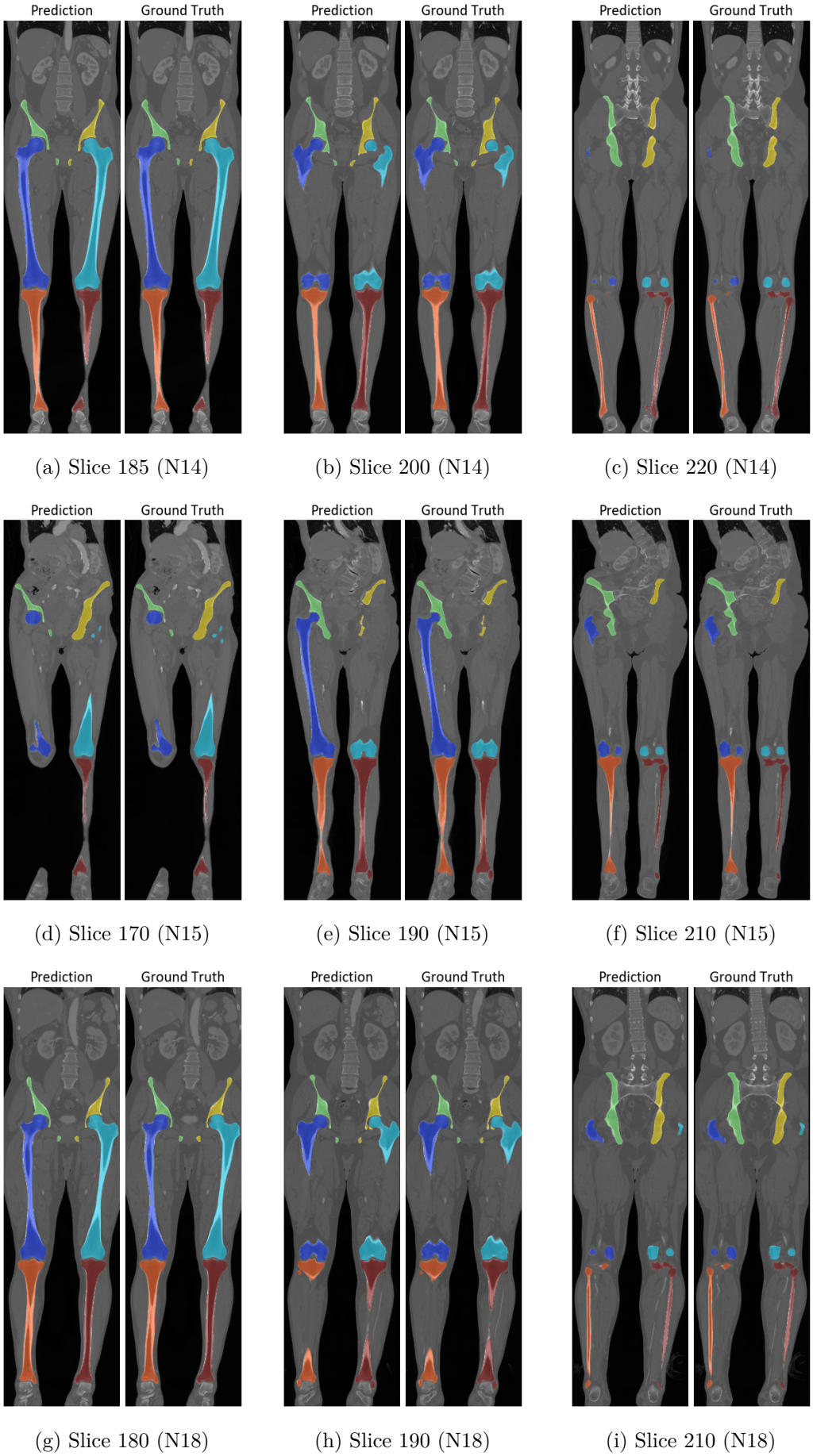
sub-images are created by cropping the image based on the bounding boxes, which are padded in each direction with 10mm. Again N12, N18 and N21 are used for visualisations. As stated above, in some cases like N18, some of the voxels far from the ROI, are predicted wrongly. Nonetheless, all predictions are accurate, even for the cases with high HDs due to these voxels. On the other hand, one single predicted voxel in the wrong location can give problems to obtain accurate bounding boxes. An example can be seen in Fig. 4.3a. Due to some of the wrongly predicted voxels, the bounding boxes of the right and left pelvis and the left tibia combined with the fibula are incorrect and result in bad sub-images. To solve this problem, before creating the bounding boxes, a median filter is applied to the prediction with a kernel size of (3,3,3), replacing each pixel in the image with the median value of its neighbouring pixels within the kernel. Such filtering gives the ability to remove these problematic predictions. Such filtering will also blur the details at the edges of the bones, but in general, the overall structures of the predictions are kept, which is the most important in this case to create the bounding boxes. Fig. 4.3b shows the updated bounding boxes and images after the median filtering. Combining the predictions with the median filtering allows great bone localisation for all of the test cases.

4.3 Bone segmentation

After the images are cropped to six sub-images, a forward pass is applied through the three trained networks for the femur, pelvis, and tibia combined with the fibula. This gives separate predictions of the bone of interest. By combining these, the bone segmentation predictions of the complete CT image are obtained. This section discusses the results of the full-resolution predictions, again combining visualisations and evaluation metrics.

The evaluation metrics are shown in Table A.5. It can be seen that the DSC, recall, and precision mean values are high for all bone classes with a low standard deviation. A full overview of the metrics for each bone class per subject can be found in A.3. The DSC, precision and recall have high mean values and low standard deviations for all classes, showing that all of the models perform well for all unseen cases. No real discrepancies are found. The lowest DSC is found for the pelvis with mean values of 0.972 and 0.974. The model that is performing best is the femur with DSCs of 0.985. This trend is also found for precision and recall.

Even though the DSC, precision, and recall are high, indicating good predictions. The HD distance seems to be high in a lot of cases. This could be because the model might not be able to capture fine details or variations in object shapes, but can again also be because of a few misclassified voxels. The highest HD values are found for the left pelvis. To investigate this matter further, visualisations have been provided in order to gain a better understanding of the predictions. Several slices of the full predictions have been included to provide an overview of the predictions, namely those for N18, which appears to be the best case, N15, which has a high



(a) Slice 185 (N14) (b) Slice 200 (N14) (c) Slice 220 (N14)

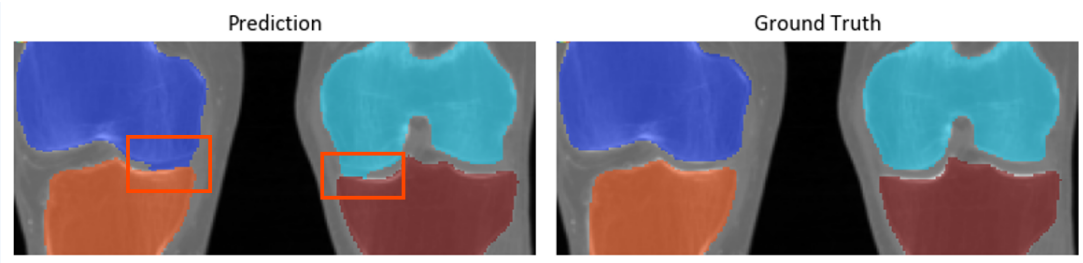
(d) Slice 170 (N15) (e) Slice 190 (N15) (f) Slice 210 (N15)

(g) Slice 180 (N18) (h) Slice 190 (N18) (i) Slice 210 (N18)

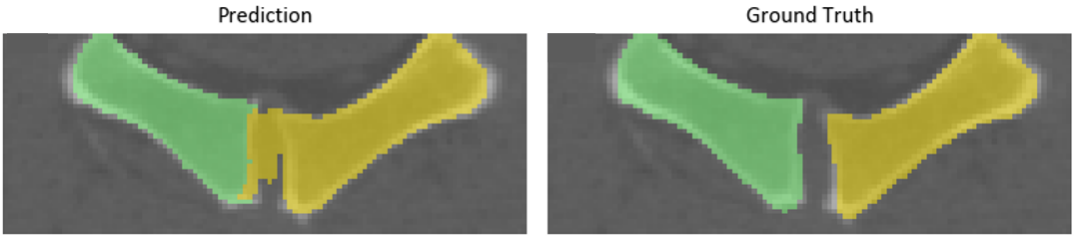
Figure 4.5: Results bone segmentation for N14, N15 and N18. Slices given in coronal direction.

Type	DSC	Precision	Recall	HD (mm)
Femur R	0.985 ± 0.002	0.989 ± 0.003	0.981 ± 0.005	3.86 ± 2.94
Femur L	0.985 ± 0.004	0.989 ± 0.003	0.981 ± 0.007	4.78 ± 2.60
Pelvis R	0.974 ± 0.002	0.979 ± 0.005	0.969 ± 0.004	7.97± 5.16
Pelvis L	0.972 ± 0.008	0.979 ± 0.009	0.965 ± 0.010	22.3 ± 26.4
Tibia + Fibula R	0.975 ± 0.006	0.979 ± 0.006	0.972 ± 0.007	5.21 ± 4.59
Tibia + Fibula L	0.977 ± 0.002	0.981 ± 0.003	0.972 ± 0.004	4.19 ± 6.60

Table 4.2: Metrics of bone segmentation predictions.



(a) Distal femur (N15)

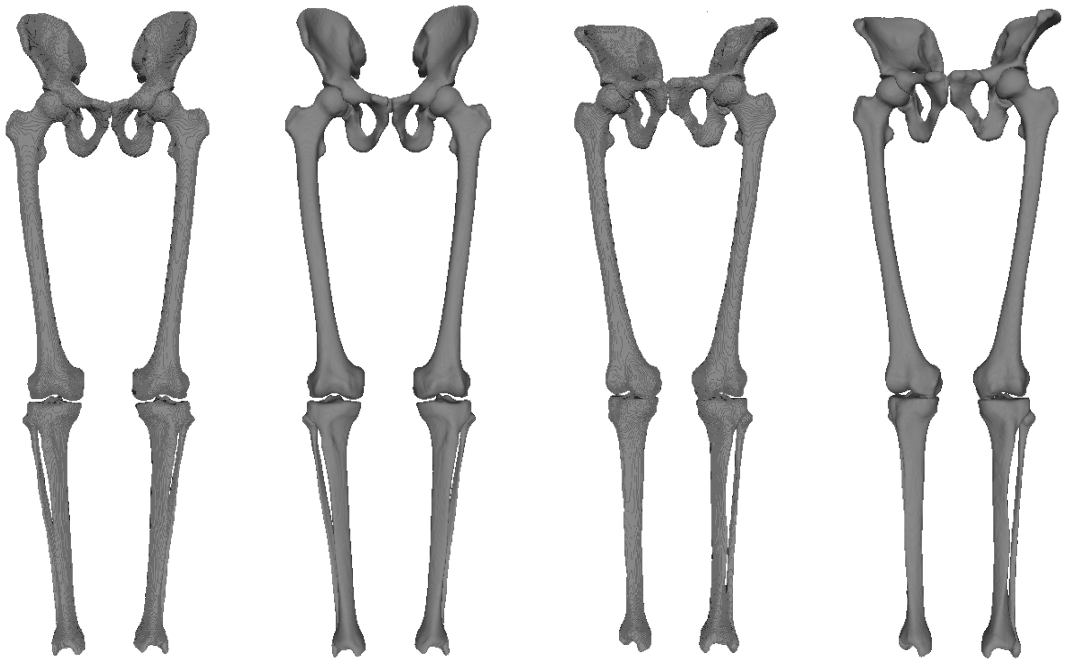


(b) Left pelvis (N14)

Figure 4.6: Examples of leakage.

HD value of 90.648 for the left pelvis, and N14, which has a lower DSC, precision, and recall value for the left pelvis when compared to the other cases. Three coronal slices are given for these cases in Fig. 4.5.

These slices confirm that the models are performing very well, giving predictions that are very close to the ground truth. The only visible difference between the predictions and ground truths can be found in slices 170 and 190 of N15. Both the right and left distal femur show some leakage towards the tibia, giving FP. A more detailed view of this leakage is shown in Fig. 4.6a. Leakage can also be found in the pelvis region for some cases. For example for N15, leakage is seen at the left pelvis, shown in Fig. 4.6b. Besides this leakage, no clear visual mistakes are found on the label maps. This also indicates that the large HDs found are probably often due to a small amount of misclassified voxels, which are difficult to spot.

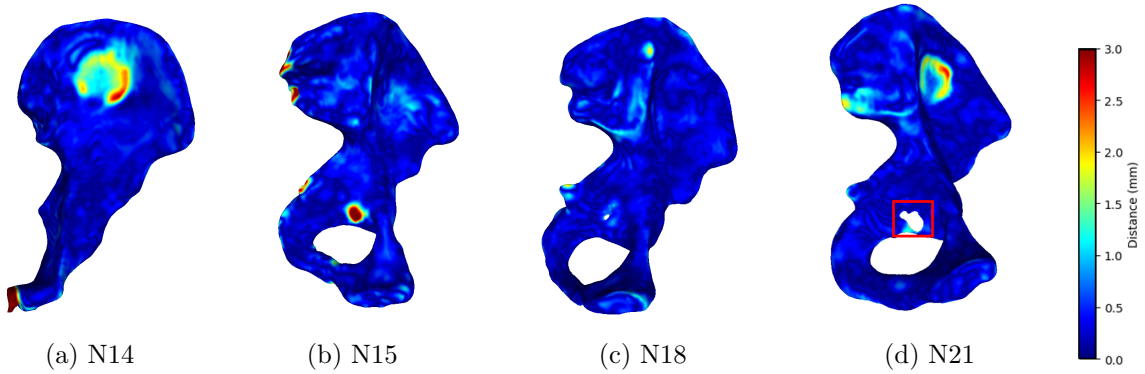


(a) Original (N14) (b) Smoothed (N14) (c) Original (N15) (d) Smoothed (N15)

Figure 4.7: 3D models of predictions.

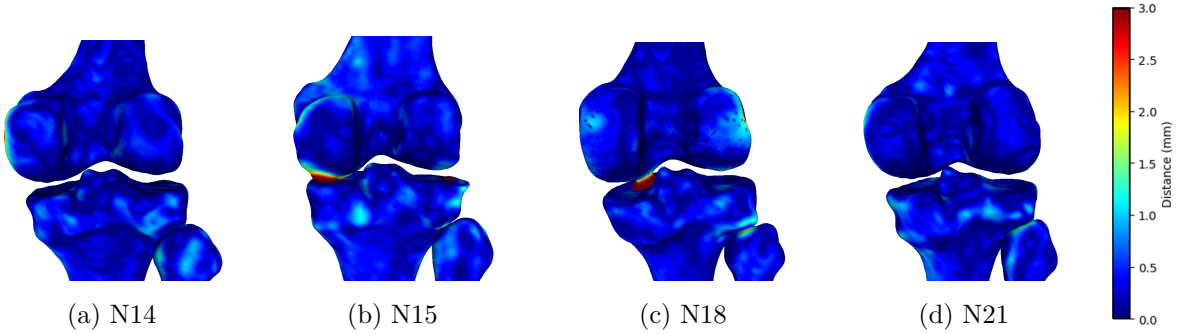
Label maps are difficult to further investigate, due to the large amount of slices. To get a better understanding of the predictions, these predicted label maps are converted to 3D models, which can give more insight into the predictions. This is done by finding the contours of the ROIs in each slice. Next, vertices are placed at the coordinates that belong to these contours, and faces are created. This is done using the marching cubes algorithm, which is implemented in Scikit-Image [55, 56]. Once 3D models are created, these are compared to the ground truth models from the original dataset. Visualisations are made in Meshlab [57]. Examples of predictions can be seen in Fig. 4.7. Note that due to the voxel-wise predictions, the created meshes are not smooth. To face this problem, some smoothing algorithms can be applied. In this work, the 3D models are further smoothed by using the Taubin smoothing available in Meshlab, also shown in Fig. 4.7 [58]. Smoothing parameters are set to $\lambda = 0.8$ and $\mu = -0.54$.

Since ground truth 3D models can be accessed, it is easier to compare the predictions with the actual results. While label masks may appear similar to the ground truth, certain errors can be identified more effectively using 3D models. To visualise these errors, the minimal distance from each vertex of the prediction to the ground truth model is calculated and used a colour-coded system to indicate the level of error. First, this analysis is performed for the left pelvis, which had the worst metrics. The results are shown in Fig. 4.8. The subjects used previously are included in the visualisation, allowing comparing the results with the label masks. These visualisations offer valuable information. For example, in subject N14, there is leakage at the



(a) N14 (b) N15 (c) N18 (d) N21

Figure 4.8: Visualisation of predictions pelvis L, coloured based on error compared to ground truth.

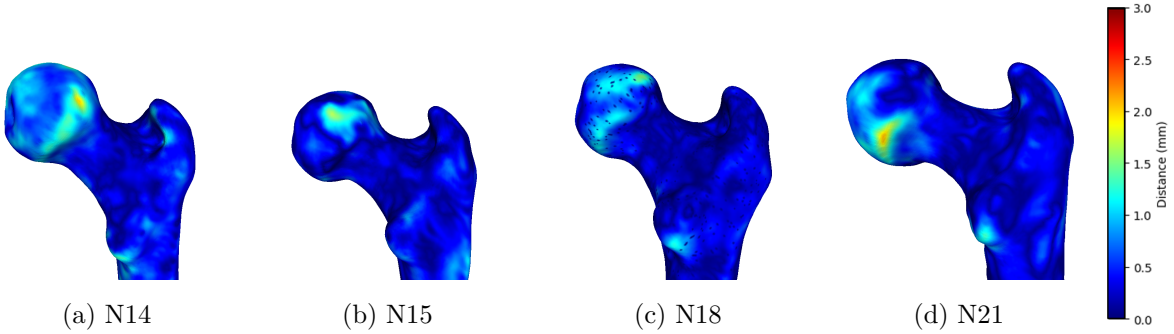


(a) N14 (b) N15 (c) N18 (d) N21

Figure 4.9: Predictions at the femur and tibia+fibula joint. Coloured based on the distance to ground truth.

left bottom, as shown before in Fig. 4.6b. Note that the orientation of the models has been changed to improve showcasing the results. In N15, several spots are wrongly predicted and differ by 3mm or more from the ground truth. However, N18 and N21 are both well-predicted, with almost no error. Nevertheless, there is a hole in the predictions for N21, as indicated by the red box and in both predictions. There is also a similarly located spot with an error of up to 3mm for both N14 and N21.

In addition to the left pelvis, the results for other bones are also visualised, including the joint of the femur and tibia and the proximal femur. As expected, the predictions for these bones exhibited lower error compared to those of the pelvis. However, leakage is still observed in the femur predictions for subjects N15 and N18. Nonetheless, the maximum distances between the predictions and the ground truth for these bones are within 1-1.5mm. Most differences remain lower than 1mm, showing that voxels at the edges are predicted perfectly as the resolution is 1mm.



(a) N14 (b) N15 (c) N18 (d) N21

Figure 4.10: Predictions at the proximal femur. Coloured based on the distance to ground truth.

4.4 Discussion

The results show that the proposed workflow has high potential. The downsampled predictions are very good, reaching mean DSCs between 0.891 and 0.949 and high recalls. On the other hand, the precision values are lower showing that the model is probably overestimating certain regions. These hypotheses are confirmed by the visualisations in Fig. 4.2, showing that the model is able to localise the bones very well, but also tends to slightly overestimate some regions, mainly the pelvis. The pelvis is also where the lowest precision is seen. A slight overestimation is not a problem for this application as the goal is to localise the correct regions and draw a bounding box around them. On the contrary, for this application, the high HDs found in certain cases are a problem. These high values indicate that certain voxels at a high distance from the ground truth region are misclassified. While these small amounts of voxels do not influence the DSC, precision, and recall much, they have a huge effect on the creation of the bounding boxes. This is for example seen in the subject N18 where a few amount of voxels lead to bounding boxes for the left and right pelvis and the left tibia combined with the fibula that are completely wrong, creating bad sub-images. To solve this, a median filter is applied to the predictions. This removes the few misclassified voxels, creating correct bounding boxes. The filtering also leads to some removal of details at the edges of the bones, but this is not a problem for the application as the bounding boxes can still be created correctly, definitely when taking the extra padding into account. Applying this first model combined with the median filtering shows to give good sub-images for all cases.

The visualisations in Fig. 4.2 confirm this hypothesis, revealing that the model can accurately localise bones, but tends to slightly overestimate some regions, particularly the pelvis, which has the lowest precision with mean values of 0.790 and 0.813. While this slight overestimation does not affect the goal of localising the correct regions and drawing bounding boxes around them, high values of HD in certain cases do pose a problem. Although the small number of misclassified

Model	Bone type	Inference time	Results
proposed model	pelvis femur tibia + fibula	15 seconds	DSC: 0.978 precision: 0.983 recall: 0.973 HD: 8.05mm
SSM-based (Audenaert et al.)	full lower limb	2 hours	surface error: 0.53mm - 0.76mm maximum error: 2.0mm - 7.8mm
UNet based (Klein et al.)	full body (myeloma)	several minutes	DSC: 0.92
V-Net based (Kuiper et al.)	full lower limb	20 minutes	DSC: 0.98 surface distance: 0.26mm
projection view and voxel group attention (Chen et al. [60])	tibia + fibula	unknown	DSC: 0.972 surface distance: 0.47mm

Table 4.3: Comparison of different large scale CT segmentation methods.

voxels does not affect DSC, precision, and recall, it can severely impact the creation of bounding boxes.

Subject N18 is a prime example of this, where a few misclassified voxels result in bounding boxes for the left and right pelvis and the left tibia combined with the fibula that are entirely incorrect, leading to bad sub-images. To address this issue, the median filter is applied to the predictions, which removes the misclassified voxels, resulting in accurate bounding boxes. While this filtering does lead to some loss of details at the edges of the bones, it does not affect the application its performance, as the bounding boxes could still be created correctly, particularly when considering the extra padding.

Most DL research for bone segmentation focuses on specific ROIs, having data of only the ROI or already cropped to this ROI before applying any DL. Examples can be found in Chen et al., and Deng et al., achieving good results using CNNs, but working on images that already focus on the ROI [10, 59]. However, in this work, a way to segment these ROIs on CTs containing the full lower limb region is found, making localisation of the ROI possible. By applying the localisation technique, large CT images can be processed more easily.

After identifying the sub-images, the remaining models are then applied to generate the final segmentation predictions. Table A.5 presents the results per case, which indicate excellent performance for all cases with mean DSCs of at least 0.972 and minimum precision and recall values of 0.979 and 0.969, respectively. Since full CT images are being used, the obtained results are compared with alternative techniques that perform segmentation on comparable data. An overview of the comparisons discussed next, can be found in Table 4.3. Note that these methods are applied on different data making them not directly comparable. First, the

method of Audenaert et al., performed on a subset of the same dataset is compared [1]. This is a SSM based method that achieves average distance errors on the pelvis, femur, tibia, and fibula of respectively 0.75 ± 0.17, 0.65 ± 0.10, 0.63 ± 0.11, and 0.76 ± 0.18. HDs on the same bones are 7.84 ± 2.26, 4.79 ± 2.39, 4.07 ± 2.15, and 3.76 ± 1.17. It could be concluded that the proposed method has higher HDs, indicating larger errors. However, it is important to note that these larger HDs are largely caused by issues such as leakage or a few misclassified voxels, rendering them not exactly suitable for comparison. The average surface distance cannot be compared as it is not included in this work. While it is true that this SSM-based method may achieve better results, it is worth noting that the inference time for an unseen case is currently around 2 hours. Thus, the proposed method is still beneficial in terms of speed.

After comparing results with a SSM-based method, comparison can be made with other DL methods. The found outcomes are superior to those obtained by Klein et al. in their whole-body CT method, which achieved DSCs of 0.92 ± 0.05 and a recall of 0.91 ± 0.08 [11]. However, it is important to note that Klein et al.'s model is designed specifically for the segmentation of patients suffering from multiple myeloma, while the methods in this thesis are created for healthy subjects, making the task less challenging. Next, the results are similar to those of Kuiper et al., reaching a DSC of 0.98 ± 0.01 on similar data as ours [12]. The found HD by Kuiper et al. is significantly lower than ours, reaching a 95th percentile HD of 0.65 ± 0.28 mm. Even though their results might be a bit better, one of the benefits of the localisation method is the fast predictions. Next to the accuracy of the models in this work, the workflow also proves to be fast. The complete workflow, from CT to a segmentation label map takes around 20 seconds. These can be converted to 3D models. The required time for this depends on the used method. 3D models of the full lower limb can be created in less than 1 min 30. In Kuiper et al. inference on new images takes 20 minutes, proving that the proposed method is a lot faster. Other studies are not included as most of them focus only on the ROI instead of obtaining the results from the full CT. While direct comparisons to other research are difficult to make due to differences in data, these findings demonstrate that results are competitive with state-of-the-art research.

Although the workflow is promising, some issues have been encountered with the final predictions, particularly when creating the 3D models. Leakage is one such problem, which occurs at the femur and pelvis, as illustrated in Figs. 4.6, 4.8 and 4.9. Additionally, the pelvis predictions are not perfect, with examples such as N21 showing a hole in the prediction and N15, N18, and N21 displaying regions where an error of up to 3mm may occur. However, the predictions for other bones are generally very accurate, with a maximum error of 1-1.5mm. The larger HD values in Table A.5 for the final predictions may be due to a few single voxels being misclassified at a larger distance from the ground truth as most predicted 3D models do not show large differences compared to the ground truth. The imperfection of the label maps could be a possible reason for the difficulties encountered and the less-than-perfect predictions. Although the ground truths are corrected manually, errors at the voxel level might persist due to factors

such as interobserver variability and the conversion from mesh to label map. The presence of errors in the ground truth at the voxel level may result in reduced accuracy and precision of the segmentation tool.

While the main goal may not be to discuss the clinical relevance of these results, it is still crucial to do so. If a tool is intended for clinical use, it must be robust and applicable to all cases, with high accuracy, as its results may significantly impact the decision-making of a clinician. Although the accuracy is currently very high and has been tested on multiple cases, it must be acknowledged that this method is not yet ready for clinical use. Issues such as leakage or challenges in difficult spots within the pelvis need to be addressed before confidently moving forward. For instance, subject-specific implants or pressure simulations based on these 3D models could potentially help mitigate the impact of any potential leakage. It is also important to note that this method has been created specifically for healthy patients, and a clinician may encounter a wider variety of patients, including those who may not have been included in this study.

The proposed workflow aims to achieve both speed and accuracy in predictions, however, the current focus is on fast predictions and there is potential for further optimisation to increase accuracy. While accuracy is not yet fully optimised, the workflow offers fast predictions that can be generated in just 15 seconds. Nevertheless, the accuracy can still be improve too to better serve users in the future. In this way, a tool that becomes clinically relevant can be created.

5 Neural flow based data augmentation method

This chapter explains a novel data augmentation technique that is able to generate new shape and image pairs that can be used as training samples for DL techniques. This technique and why it has been developed is further explained in Section 5.1. Section 5.2 explains an existing data augmentation method and its shortcomings. Next, Section 5.3 presents a novel data augmentation method, highlighting the FlowSSM and its application. While in Section 5.3 the general background and method is explained, in Section 5.4 the workflow is demonstrated on two use cases.

5.1 Introduction

The created worfklow using the UNet seems to give very accurate results in the case of lower limb segmentation of the bones. This can be traced back to the fact that CT data is used, wherein the bone exhibits a high contrast in comparison to the surrounding tissue. Only in the joints regions, where contrast between the bone and other structures is lower, the UNet has more difficulties in accurate segmentation. Remaining difficulties should be addressed through alternative techniques instead of increasing the dataset.

Other deep learning models that focus on more difficult anatomical structures with lower tissue contrast might need more labelled data to be able to achieve accurate results. Moreover, CNN-based models like DeepSSM that directly construct shapes from images by predicting low-dimensional shape descriptors require a substantial amount of data [48]. To overcome this limitation, a new method to create training samples is proposed. This method consists of pairing a 3D shape model of an anatomical structure with a 3D medical image, thereby generating new training data. Generated training samples must follow the population statistics of the original data to create representative training samples. This way, the performance of deep learning models can be improved, especially when there is a clear need for more labelled data. While this

thesis focuses specifically on the application of this data augmentation technique on CT images and bone structures, it is important to note that this technique is not limited to this application. It can be used for almost any anatomical structure for which corresponding medical images are available.

This method was developed under the assumption that the UNet would require a larger dataset to generate accurate predictions. Therefore, further data augmentation for the segmentation of the lower limbs was not pursued. However, the novel technique is still included in this thesis as it may benefit other applications.

Data augmentation to generate new training samples has been previously proposed in the literature, such as in the case of DeepSSM. Nevertheless, the method from the authors of DeepSSM has some limitations that need to be addressed. To overcome these limitations, a newly proposed workflow is introduced in this chapter. The first step of this workflow involves generating shape models using a neural flow method called FlowSSM [61]. This method allows for the generation of 3D shape models that adhere to the population statistics. Once the shape models have been generated, the next step is to generate corresponding medical images.

5.2 DeepSSM data augmentation method

The proposed method builds upon the data augmentation approach introduced by the authors of DeepSSM. Hence, the first data augmentation by DeepSSM is discussed. The workflow consists of embedding shapes in a lower-dimensional subspace, sampling to generate new shapes, and generating images from these shapes. However, there are limitations to this approach, particularly in terms of the embedding of shapes in the lower-dimensional space. Hence, the novel method mainly focuses on improving this part. In the following parts, each step of the DeepSSM data augmentation is explained in detail starting with the method to embed shapes in the lower-dimensional subspace, highlighting the disadvantages. Next, the sampling process is explained, followed by a detailed description of the image generation step. Addressing the limitations of the DeepSSM approach will make it clear that a novel data augmentation method is necessary.

5.2.1 Shape embedding and generation

The first step in the DeepSSM data augmentation pipeline is to embed the original shapes in a lower-dimensional subspace using PCA. To complete this task, the ideas of statistical shape models are used. Each shape can be represented as a point cloud, a $\mathbb{R}^{n\times 3}$ vector representing n 3D coordinates. The shapes are denoted by $C = \{C_1, ..., C_N\}$. Note that the techniques that

follow are not limited to only this kind of representation, but can be adapted for other shape representations. Next, these representations are collected into a matrix, which is then subjected to PCA to reduce the dimensionality. Each shape can now be represented as a PCA loading in $\mathbb{R}^M$, from which the original shape can be reconstructed through inverse PCA, applying Eq. 2.9. The loadings are denoted by $Z = \{Z_1, ..., Z_N\}$ and create the PCA space. The dimension M of the lower-dimensional PCA space is determined by the number of eigenmodes selected, which depends on the amount of variance the user wishes to describe, explained in Subsection 2.5.2.

Once Z is found, it can be modelled by a chosen distribution, such as multivariate Gaussian or Kernel Density Estimation (KDE). This modeling allows for sampling from Z to generate new PCA loadings. Applying inverse PCA on the generated loadings generates new, augmented shapes that follow the population statistics modelled by the distribution. This way, thousands of new shapes can be created.

While the proposed shape augmentation technique has shown promising results, it has certain limitations. One such limitation is that PCA can only represent linear data, while variation in certain anatomical structures, like skeletal shapes, is inherently nonlinear. This means that certain nonlinear features inherent to the structure cannot be generated. Another limitation is the requirement for the alignment of the shapes before applying PCA. This means that before applying PCA, each shape is aligned from its local coordinate system to a world coordinate system. This is done by making use of a rigid transformation method, making use of rotations and translations. In practice, corresponding shapes can be aligned GPA. This alignment ensures that the resulting PCA loadings accurately capture the underlying variability in the shapes. Because all of the original shapes are aligned when creating the PCA space, the generated shapes will also follow the same alignment.

It is also important to note that the used version of PCA needs correspondence between the point clouds C. Because not every dataset has correspondences available, DeepSSM does provide a way of creating these. Nevertheless, this method remains difficult to optimise.

Finally, it should be noted that DeepSSM suggests using a TL network encoder to embed shapes into latent embeddings. However, this encoder was not utilised for data augmentation and would require the use of an extra decoder component.

5.2.2 Image generation

To enhance the performance of deep learning methods using new shape-image samples, a method for obtaining the images corresponding to the generated shapes is needed. This can be achieved by following a specific workflow for each generated shape. Firstly, the closest shape in the original population is identified in the shape space. Then, a thin spline plate (TSP) warp is

calculated from the closest shape to the generated shape. Finally, the warp is applied to the image corresponding to the closest shape. By repeating these steps for each generated shape, numerous shape-image samples can be generated, providing valuable data for training deep learning models.

The TSP warp is a mathematical technique to find a continuous deformation between two shapes [62]. The transformation is represented by a thin spline plate, which is a flexible surface that can be deformed to match the shape of the target object. The warp is defined by a set of control points on the thin spline plate that are adjusted to minimise the distance between corresponding points on the two shapes. It is important to note that this technique requires correspondence. The problem can also be stated as follows: the goal is to find a smooth function $f : \mathbb{R}^3 \to \mathbb{R}^3$ that maps a 3D coordinate of the original shape to the target coordinate of the target shape. There is only a sparse amount of control points, specifically the available vertices. Therefore this could also be seen as an interpolation problem: find the smooth function f, given the set of control points.

To further explain the technique, this interpolation problem is solved in 1D, showing the general ideas [63]. These can be easily expanded to 3D. A visualisation of the 1D method can be found in Fig. 5.1. Imagine that N control points $CP = \{(x_1, y_1), ..., (x_N, y_N)\}$ are available of a smooth function $f_1D : \mathbb{R} \to \mathbb{R}$ in which $f(x_i) = y_i$. As we're trying to find the function f_1D, the only available information is the control points. The solution lies in the use of radial basis functions (RBFs), a type of function that produces a value based on the distance to a preset point. An example of an RBF in 1D is a Gaussian kernel $e^{\frac{-r^2}{2\sigma}}$. By placing an RBF at each control point x_i, $f(x)$ can be rewritten as:

$$f(x) = \sum_{i=1}^{N} \alpha_i RBF(x, x_i), \tag{5.1}$$

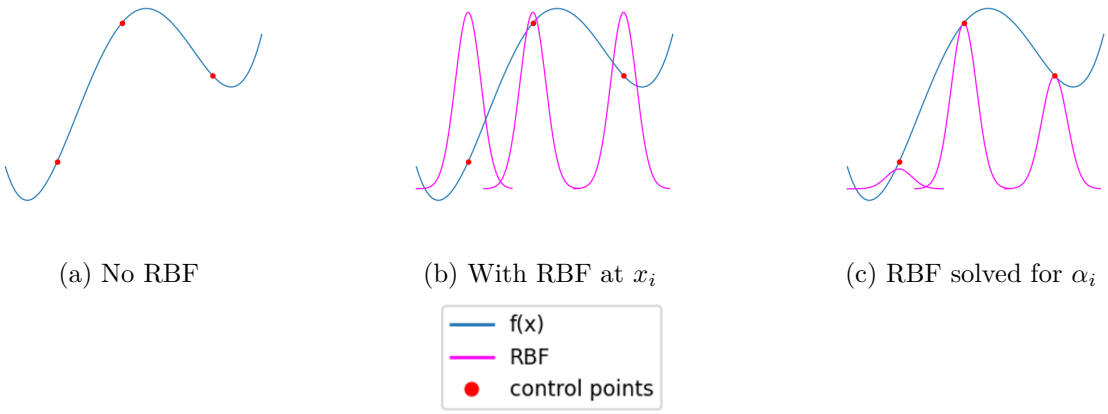
where $RBF(x, x_i)$ is the RBF at x corresponding to control point x_i and α_i the weight corresponding to the RBF. The values of α_i define f_1D. In the case of CP, the values of α_i can be found by solving the following linear system:

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} RBF(x_1, x_1) & \dots & RBF(x_1, x_N) \\ \vdots & \ddots & \vdots \\ RBF(x_N, x_1) & \dots & RBF(x_N, x_N) \end{bmatrix}}_{\mathbf{RBF}} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}}_{\boldsymbol{\alpha}} \tag{5.2}$$

A solution can be found by applying:

$$\boldsymbol{\alpha} = \mathbf{RBF}^{-1}\,\mathbf{y}. \tag{5.3}$$

Expanding these equations in 3D gives an idea of how a TSP warp can be applied on the control points, the vertices. Which RBFs are exactly used in DeepSSM is not clear. Once the TSP warp is found, it can easily be applied to the images giving the final shape and image pair sample.



(a) No RBF (b) With RBF at x_i (c) RBF solved for α_i

Figure 5.1: Visualisation of the RBF method in 1D.

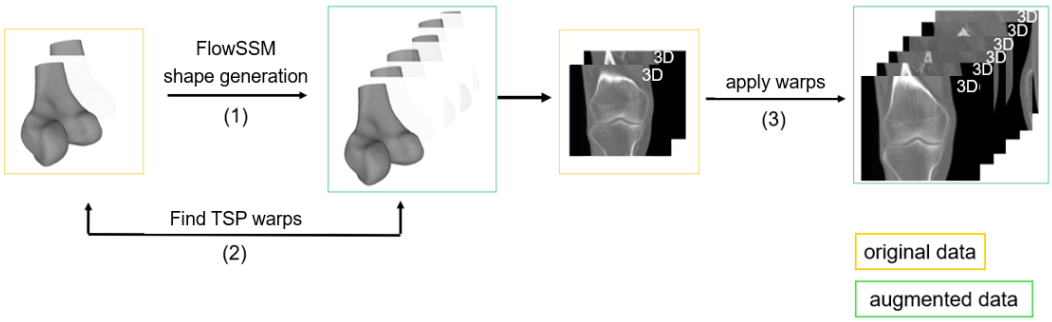


Figure 5.2: Data augmentation workflow.

The primary limitation in the use of the produced images is the alignment issue, as discussed previously. Due to the shape alignment method utilised needed for PCA, the images maintain this alignment too. This can pose a problem when using the images for some deep learning techniques, as changes in shape orientation can significantly affect performance. To remedy this, test images must undergo alignment with the training set using image registration techniques. However, this process can be more challenging than the segmentation process itself.

5.3 Neural flow-based data augmentation method

To address the limitations of the need for alignment and the inability to fully capture nonlinear shape variations, due to the use of PCA in DeepSSM, a novel method is proposed. An overview of the complete proposed method is shown in Fig. 5.2. The main difference between the proposed method and the DeepSSM data augmentation is the used shape generation method. A different way of generating shapes is used, able to deal with nonlinearities and without the

need for alignment. This is done by using FlowSSM, allowing for the generation of shapes using a different approach than sampling the PCA space. The next step is finding the warp of each generated mesh to its closest training mesh. Finally, the same warps are applied to the training image corresponding to the closest training mesh. As the novelty of this method lies in using FlowSSM for shape generation, this part mainly focuses on explaining this technique and how it can be used for shape generation. The proposed method offers a novel way to generate shape-image samples that can deal with shape nonlinearities and do not require alignment. This allows for a more representative and diverse dataset, which can improve the performance of deep learning techniques. First, FlowSSM is explained in more detail. Next, the generation of shapes using FlowSSM is discussed. Note that the image generation of the novel method is equal to that of DeepSSM and will not be further explained in much detail.

5.3.1 FlowSSM

FlowSSM is a cutting-edge technique aimed at creating a neural flow-based SSM that accurately models natural shape variation [61]. Unlike the classical SSM that relies on PCA and requires correspondences, FlowSSM does not require any correspondences. Instead, it utilises multi-layer perceptrons (MLPs) to capture the neural flow deformation of the shapes. This results in obtaining latent shape representations for each shape, which enables describing the population in the latent space. This innovative technique enables the generation of thousands of new shapes that adhere to population statistics.

In FlowSSM, a particular type of shape is modelled by training a decoder that can generate a continuous flow to deform a template shape into a target shape. This is achieved by learning continuous deformations of the template surface to the target. During training, not only the flow is learned by the network. Latent representations are also optimised together with the decoder, leading to a latent space that accurately represents the population. This is called an implicit decoder [59].

The template surface $\mathcal{T}$, which can be any shape of choice, is deformed to target surface X by applying a continuous deformation on each surface point $x_0 \in \mathcal{T}$. $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^3$ is denoted as the continuous trajectory where $\mathbf{x}(0) = x_0$ and $\mathbf{x}(T)$ is the deformed point, corresponding to the target surface. Next, $\mathbf{v} : \mathbb{R}^3$ x $[0, T] \rightarrow \mathbb{R}^3$ models the velocity field depicting the change of $\mathbf{x}$ in time and space. The following ODE describes the trajectories based on $\mathbf{x}$ and $\mathbf{v}$:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}(\mathbf{x}(t), t) \qquad t \in [0, T]. \tag{5.4}$$

Solving this equation with the initial condition $\mathbf{x}(0) = x_0$ gives the deformation flow $\Phi : \mathbb{R}^3 \times [0, T] \rightarrow \mathbb{R}^3$ that describes the full process to obtain the target surface, satisfying

$$\frac{d\Phi}{dt} = v(\Phi(x_0, t), t) \qquad \Phi(x_0, 0) = x_0. \tag{5.5}$$

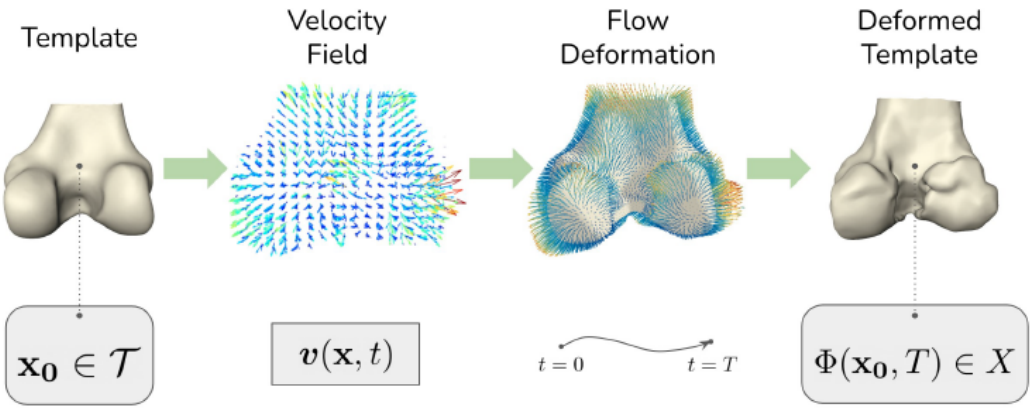


Figure 5.3: Flow deformation overview [64].

A visualisation of this can be found in Fig. 5.3 [64].

As mentioned, the deformation flow is captured by an implicit decoder network. The network used is an IM-Net [59], which will be shortly explained. The input of the network is a coordinate together with a parametrisation of the target shape. The output of the network is the displacement vector at time point t. The parametrisation is important for the application of shape generation using FlowSSM. This is a latent embedding of the target shape and remains the same for different values of $t \in [0, T]$. There are two kinds of shape parametrisations used in FlowSSM, global and local. The global one is a latent vector $z_g \in \mathbb{R}^d$, where d can be set before training. However, only using a global parametrisation leads to smooth and low-frequency deformations. In Gupta et al. for example, the use of only a global parametrisation seemed to produce over-smooth meshes [65]. Therefore, a local parametrisation $z_l \in \mathbb{R}^{d \times 3k}$ is included. As two kinds of parametrisation exist, also two networks or deformers are used. These are applied in sequence. First, a global deformer, using global parametrisation, is used to do low-frequency deformation. Next, a local deformer uses the local parametrisation to create high-frequency deformations.

5.3.2 Training deformers

Now that FlowSSM and its functioning are explained, the training of the deformers is explained in further detail. The IM-Net used is a network consisting of 5 fully connected layers with skip layer connections and leaky ReLu activation functions. An example of this IM-Net for a latent vector of $z \in \mathbb{R}^{128}$ is shown in Fig. 5.4 [64]. The input is a spatial coordinate at time-point t and the shape specific latent vector. This leads to a displacement vector at time-point t as output.

In the first half of the epochs, the global deformer and the global latent embeddings are optimised. The second half optimises the local deformer and the local latent embeddings. During

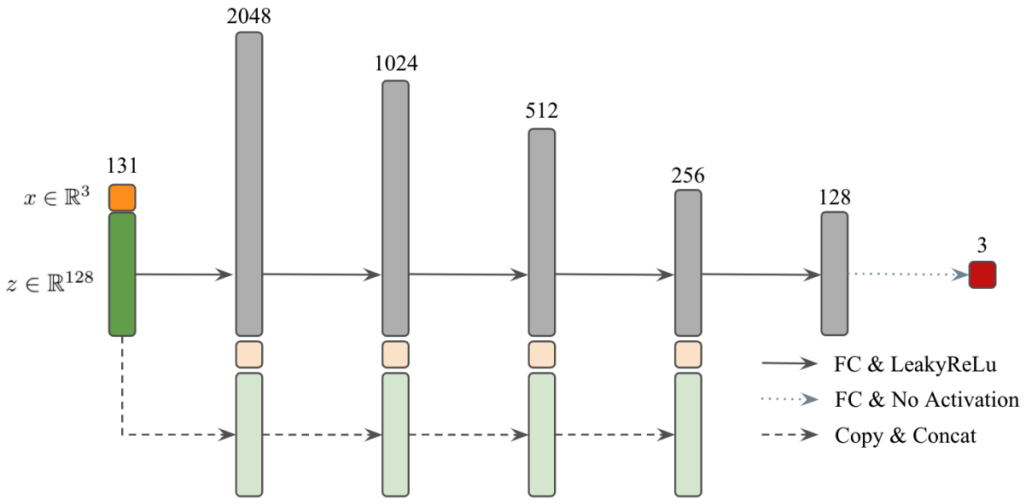


Figure 5.4: IM-Net overview for a latent vector $z \in \mathbb{R}^{128}$ [64].

training the symmetric point-set to point-set Chamfer distance (CD) is minimised, given by:

$$CD(P_i, P_\Phi) = \frac{0.5}{2|P_i|} \sum_{x_i \in P_i} \min_{x \in P_\Phi} \|x_i - x\|_2 + \frac{0.5}{2|P_\Phi|} \sum_{x \in P_\Phi} \min_{x_i \in P_i} \|x_i - x\|_2 \tag{5.6}$$

calculated between sampled surface points of the target $P_i \subset X_i$ and the corresponding deformed surface point of the template P_Φ. The deformation flow is found by minimising this over the complete training data.

Even though the framework containing this complete training procedure has been made available by FlowSSM, it contains a high amount of hyperparameters that need to be set before training: the number of epochs, learning rate, dimensions of the latent embeddings (global and local), batch size and more. Hence, finding an optimal combination of these hyperparameters is not an easy task. Luckily FlowSSM proposes a set of hyperparameters for a few example cases. Nevertheless, tuning these for a new case remains a difficult challenge. In this work, most of the parameters are fixed, and the effect of the size of the latent embeddings is further explored.

To check if the deformers are able to generalise well, it is important to also see if the template is able to deform to unseen cases. Therefore, the latent embeddings of the unseen case need to be found. This is done by only optimising the latent embeddings and keeping the deformers fixed, by minimising the Chamfer distance. The latent embeddings are randomly initialised from a normal distribution with a mean of 0 and a standard deviation of 0.01: $\mathcal{N}(0, 0.01)$. Also here the global representations are optimised before the local ones. To optimise these latent embeddings, an extra hyperparameter needs to be set, the number of optimisation epochs.

5.3.3 Shape and image generation

Once the deformers and latent embeddings of the training set are optimised, these can be used to generate shapes. This part explains how shapes can be generated. This method is also implemented by FlowSSM but is mostly used to measure the specificity of the model, being the ability to generate shapes that belong to the training population.

In this work, these generated shapes are utilised for the novel augmentation method. FlowSSM allows the generation of thousands of shapes by making use of the created latent spaces during training. Both the latent embeddings of the global and local deformers are optimised. This gives a global latent space S_g and a local latent space S_i, which is fixed after training. By sampling from S_g and S_i, new latent embeddings can be created. By deforming the template using these embeddings, new shapes are created. Therefore it is important to know how this sampling happens exactly.

Sampling happens separately for the global and local embeddings. It is also possible to apply one sampling step to obtain both together, but this technique is not further investigated. First, PCA to the complete space gives the mean and the main modes. For each shape that is generated, random weights are generated from $\mathcal{N}(0, 1)$. The amount of generated weights is equal to the number of modes used, which is set to the maximum being the number of training shapes minus 1. Next, Eq. 2.10 is applied where b_t is equal to the generated weights from the normal distribution, giving a new embedding. The same procedure is followed for S_g and S_i. Providing these latent embeddings to the deformers generates the shapes.

Once shapes are generated, they are used to generate the corresponding CT images. This is done in the same way as in DeepSSM using the TSP warp. Note that using the TSP warp requires corresponding shapes. Even though FlowSSM does not require correspondence of the shapes, completing the data augmentation with the TSP warp does.

5.4 Application on distal femur and pelvis

The previous sections provide the required technical details to apply the novel data augmentation method to different use cases. The first use case is the distal femur, as FlowSSM already showed that their method functions on the distal femur. After this, the same workflow is applied on a more difficult case, the pelvis. This section explains the workflow to obtain shape and image pairs of these cases. The used dataset is the same as presented in Chapter 3. In this part, the meshes and CT images of the training set are worked with, containing 72 samples. Both the left and right sides are used.

5.4.1 Data preprocessing and template shape

Before training the deformers, the meshes and images need some processing. For the first use case, only the distal femur is considered, and therefore, the available meshes of the femur are first cropped. First, a single mesh is cropped by hand. This reduces the number of vertices to 6151 and the number of faces to 12223. Next, each mesh is automatically cropped keeping the corresponding vertices and faces. In this way, correspondence between all meshes is kept. For the pelvis, no cropping is applied. Note that the right and left sides of the pelvis are already separated giving a total of 25967 vertices and 51934 faces for each mesh.

The next step in the preprocessing, or first step in the case of the pelvis, is the centring of the meshes. This is done by translating each mesh so that its mean coordinate is brought to (0,0,0). As FlowSSM deforms a template, the meshes in their original position cannot be used. Otherwise, the needed deformation is too large making it more difficult to achieve good results. The same translation is applied to the CT images by translating their origin. This way, the meshes still match the CT images. It is important that the meshes are centred, but not further aligned. A data augmentation technique is desired to overcome the DeepSSM limitation of the need for alignment. Therefore, further alignment is not applied, even though FlowSSM did this as an extra preprocessing step. It is assumed that the FlowSSM technique is capable of learning the orientation and can implement it in the latent embeddings.

After the translation, the images are cropped to the ROI. This is done by creating a general bounding box of all meshes and cropping the images based on this bounding box, giving the same image size for each sample. In the case of the distal femur, it is made sure that the upper part of the image is equal to the bounding box of the corresponding mesh and that only the bottom part is expanded to match the image size. The meshes and CT images of the left side are mirrored, giving 144 shape and image pairs in total.

Once these preprocessing steps are completed, the required shape and image samples are available to apply the novel training augmentation method. FlowSSM is applied as the first step, which requires a template shape. Therefore, there is a need for defining this template shape. Although FlowSSM provides a technique for creating a template shape, it is not their main focus. Their method is designed for noncorresponding meshes and is time-consuming to calculate. Since correspondence is available and is needed for the TSP warp, a different approach is taken to define the template shape. For the distal femur, the shape closest to the mean training shape is selected, while for the pelvis, the mean training shape is used.

Parameter	Distal femur	pelvis
train epochs	100	300
train lr	0.001	0.001
embedding epochs	250	350
embedding lr	0.01	0.01
batch size	8	4
d	[4,8,16,32,64]	128
k	7	6

Table 5.1: Labels and their corresponding structure.

5.4.2 Hyperparameters

Due to the high amount of hyperparameters, it is difficult to find the optimal combination without spending a large amount of time. Hence, most of the hyperparameters were set on a fixed value, based on the FlowSSM documentation. For the distal femur, the effect of the size of the latent embeddings is investigated. The exact values can be found in Table 5.1.

5.4.3 TSP warp

The template shape is in correspondence with the rest of the shapes. Hence, the generated shapes, i.e. deformed template shapes, are also in correspondence with the rest of the data. This allows to find a TSP warp between each generated shape and the closest shape in the training set. Once such a warp is found, it can be applied to the image corresponding to the closest shape. The closest training shape $X \in TS$ to a generated shape G is found using:

$$\underset{X \in TS}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{v}_{G,i} - \mathbf{v}_{X,i} \right\|_2 , \tag{5.7}$$

with $\mathbf{v}_{X,i}$ depicting a vertex of shape X and n the total number of vertices.

The TSP warp is found and applied to the image using ShapeWorks. Finding the warp between shapes with a large number of vertices can be time-consuming. Therefore, to save time, the warp is found between a smaller set of randomly sampled vertices. This way, the calculation time is not too much. Using fewer vertices than the available amount can lead to a small loss of detail. For the distal femur, 1000 sampled vertices are used. Results of the complete procedure will be shown in Chapter 6.

5.5 Conclusion

This chapter proposes a novel data augmentation technique making use of FlowSSM to generate new shapes that adhere to population statistics. This is combined with the TSP warp proposed by the authors of DeepSSM. With this data augmentation, thousands of new training samples can be generated, each consisting of a shape and an image. These can be used to train DL techniques that require a lot of data. The workflow to apply this on two use cases is also discussed. Results of these use cases can be found in Chapter 6.

6 Neural flow based data augmentation results

This chapter contains the experimental results of the neural-flow based data augmentation explained in Chapter 5. As this workflow consists of 2 major parts, the use of FlowSSM and the warping, these are explained separately in further detail. Section 6.1 explains the results of applying FlowSSM for the distal femur and the pelvis in further detail. The warping, leading to the generated images, is shown in Section 6.2. Final results are further discussed in Section 6.3.

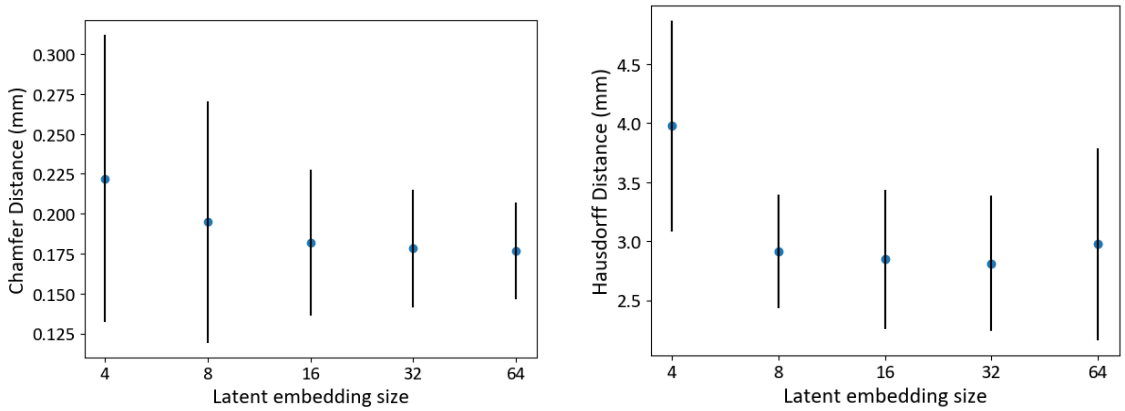
6.1 FlowSSM results

The first step in the neural flow-based data augmentation method is the generation of new shapes that adhere to the population statistics. This is done using FlowSSM. This section discusses the results of FlowSSM for both the distal femur and the pelvis. The deformers are trained as depicted in Section 5.4. The training process for both the distal femur, including shape generation, and the pelvis requires approximately 7 hours and 24 hours respectively. This is because of the difference in the number of vertices and the increase in number of epochs. For the distal femur, the effect of the size of the latent embeddings is further investigated by checking the ability of the trained deformers to generalise well when changing the size of the latent embedding. Next to the ability to generalise well, this section will also investigate the distribution of the latent embeddings and the generated shapes in more detail.

6.1.1 Generalisability

Distal femur

To evaluate the ability of the trained deformers to generalise on unseen cases, the latent embeddings of these cases are optimised and the template deformed accordingly, as detailed in



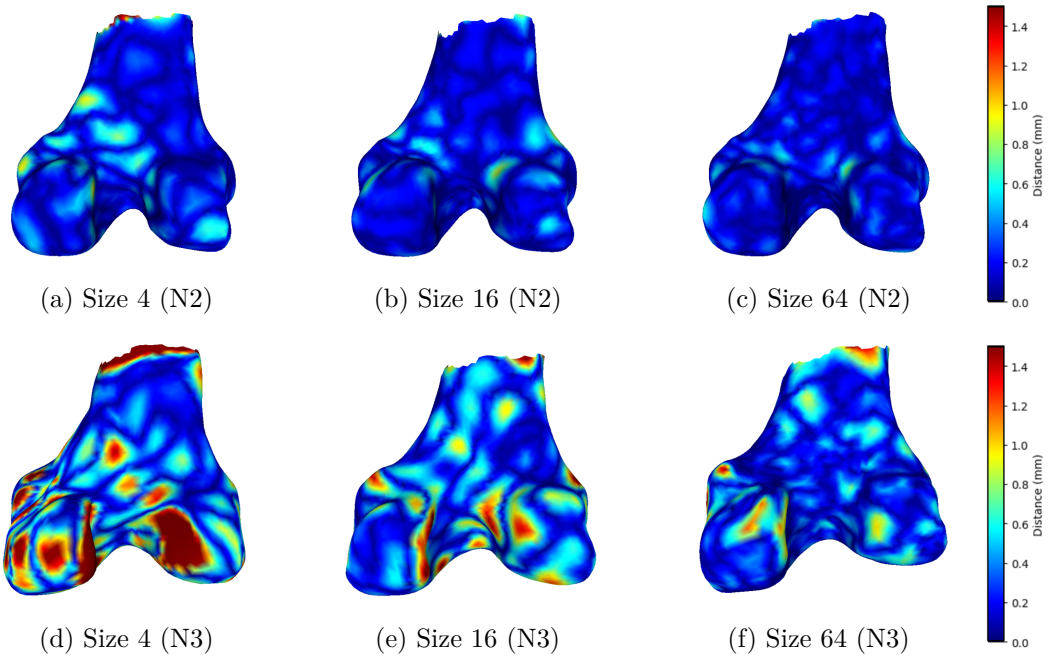
(a) Chamfer distance

(b) Hausdorff distance

Figure 6.1: Chamfer and Hausdorff distance for different latent embedding sizes to show generalisability of the deformers.

Subsection 5.3.2. Specifically, to assess the performance of the deformers, the CD of Eq. (5.6) and the HD of Eqs. (3.8)-(3.9) between the ground truth shape and the deformed template, on each validation case are calculated. Note that errors can stem from either a suboptimal latent embedding or poor generalisation of the deformer. Fig. 6.1 presents the CD and HD metrics for latent embedding sizes ranging from 4 to 64. It is worth mentioning that, in the case of the distal femur, a size of d represents a combination of a global embedding of size d and a local embedding of size $(d, 7, 7, 7)$. For each size, the mean and standard deviation is reported. This analysis indicates that for the distal femur, the optimal size of the latent embedding is between 8 and 64, as these sizes exhibit very similar metrics. However, a size of 64 results in a minor increase in HD and a minor decrease in CD. A size of 4 leads to a noticeable increase in HD and has a higher mean with a larger standard deviation regarding the CD too. Hence, based on these plots, it can be concluded that a latent embedding size of 4 is too small. These plots not only facilitate comparison between different sizes of the latent embeddings but also provide insights into the generalisability of the deformers. Overall, the findings suggest that the deformers are capable of generalising well on the distal femur data. Specifically, the CDs for sizes 8,16,32, and 64 are 0.222±0.09mm, 0.195±0.08mm, 0.182±0.05mm and 0.177±0.03mm, respectively, indicating that the average error between the deformations and ground truths is within an acceptable range of 0.2mm.

To test the hypothesis that a latent embedding size of 4 is too small and that other sizes generalise better, visual representations of deformations were created and colour-coded based on their deviation from the ground truth. The methodology outlined in Chapter 4 was followed. The focus was on the right femur of N2 and N3 since they displayed the lowest and highest CD and HD values, regardless of the embedding size. This allowed for the illustration of the best and worst-case scenarios. Visualisations for embedding sizes of 4, 16, and 64 are provided, as



(a) Size 4 (N2) (b) Size 16 (N2) (c) Size 64 (N2)

(d) Size 4 (N3) (e) Size 16 (N3) (f) Size 64 (N3)

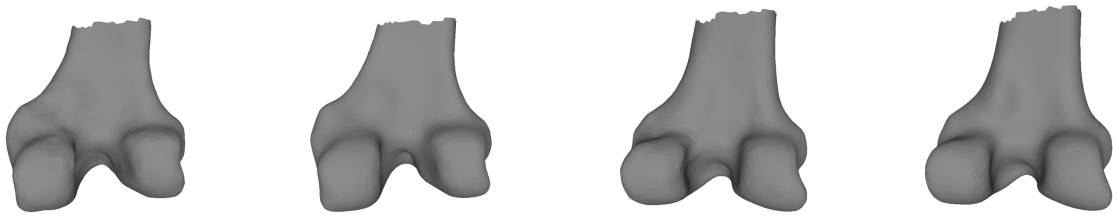
Figure 6.2: Deformations of the right femur of N2 and N3 with different sizes of latent embedding.

shown in Fig. 6.2.

The findings indicate that the generalisation of a size 4 embedding is not sufficient. For both N2 and N3, the deformed template deviated significantly from the ground truth compared to other embedding sizes. Although initially minor differences were expected between sizes 8 and 64 based on Fig. 6.1, the visualisations showed a clear improvement when increasing the latent embedding size to 64. For N2, the reconstruction was almost perfect, and the error for N3, the worst-case scenario, was also acceptable.

One of the most important questions is whether FlowSSM can learn the orientation of shapes without any alignment applied as preprocessing. The shapes are only centred, with no further alignment applied. Based on the previous measurements of HD and CD and the visualisations created, it can be assumed that the orientation does not pose any problem and is learned by the deformers. To confirm this, two cases from the validation set are presented, and their deformation is compared to the ground truth in Fig. 6.3. These cases have a clear difference in orientation, making them ideal to test this hypothesis. They are the right distal femur of N1 and the left, mirrored, distal femur of N9. Based on these cases, it can be concluded that the deformers are able to learn the orientation almost perfectly. This is also apparent in all other cases.

After reviewing the results, it can be inferred that FlowSSM performs exceptionally well in



(a) Deformed (N1) (b) Ground truth (N1) (c) Deformed (N9) (d) Ground truth (N9)

Figure 6.3: Deformations compared with the ground truth for N1 (R) and N9 (L). This shows the ability of the deformers to learn the orientation.

generalising the dataset for the distal femur. The deformers exhibit a remarkable ability to shape the template to match the target form using only its latent embedding, even in cases that are previously unseen. This also suggests that the latent space has been constructed correctly, and can effectively capture the underlying structure of the data.

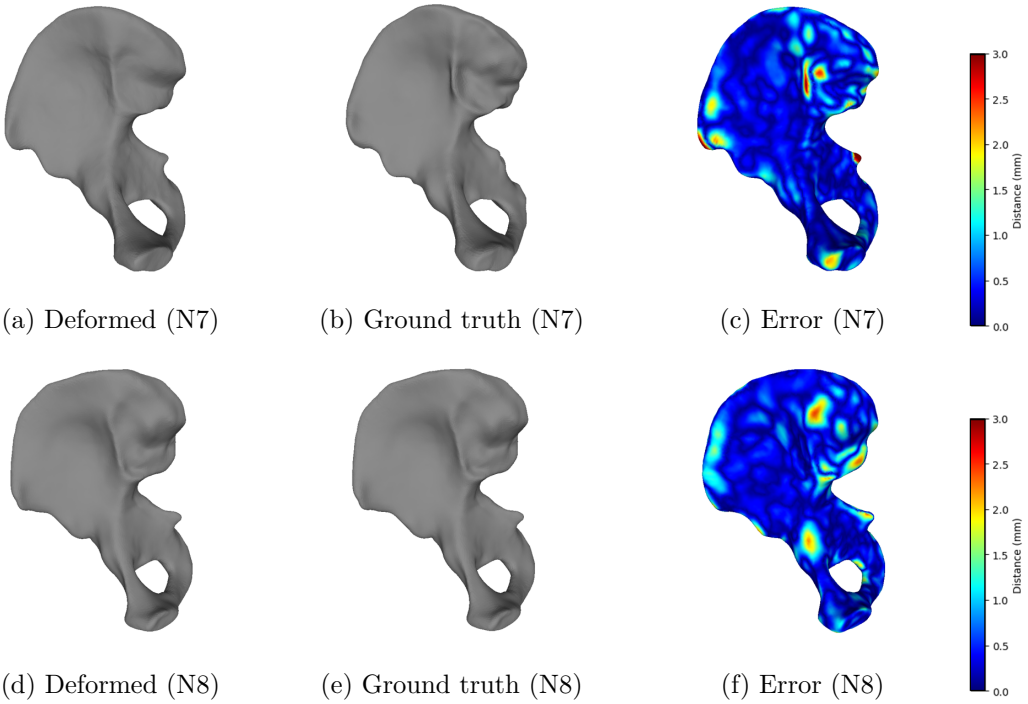
Pelvis

The latent embedding is not further investigated for the pelvis. Hyperparameters are chosen based on available documentation. Details about the chosen hyperparameters can be found in Table 5.1. When evaluating the generalisability of the model using CD and HD, values of 0.477 $\pm$ 0.04mm and 4.688 $\pm$ 0.90mm are obtained. Although these values are higher than those for the femur case, a mean CD of 0.477mm indicates that the model is still able to generalise well on unseen cases.

To further assess the model its generalisability, the deformations with the smallest and largest HD are visualised, which were found to be 2.869mm and 6.199mm, respectively. These corresponded to the left, mirrored pelvis of N8 and the right pelvis of N7. The resulting deformations were compared to their respective ground truths and coloured based on their error. As shown in Fig. 6.4, the deformations matched the ground truths, demonstrating the model its ability to generalise. Although some areas showed larger errors, the overall shape was reconstructed very well. Again, it can also be seen that the model is able to learn the orientation close to perfection.

6.1.2 Generated shapes

After training the deformers and showing the ability to generalise well over the dataset, FlowSSM is used to generate shapes as explained in Subsection 5.3.3. With each training, 1000 shapes are generated.



(a) Deformed (N7) (b) Ground truth (N7) (c) Error (N7)

(d) Deformed (N8) (e) Ground truth (N8) (f) Error (N8)

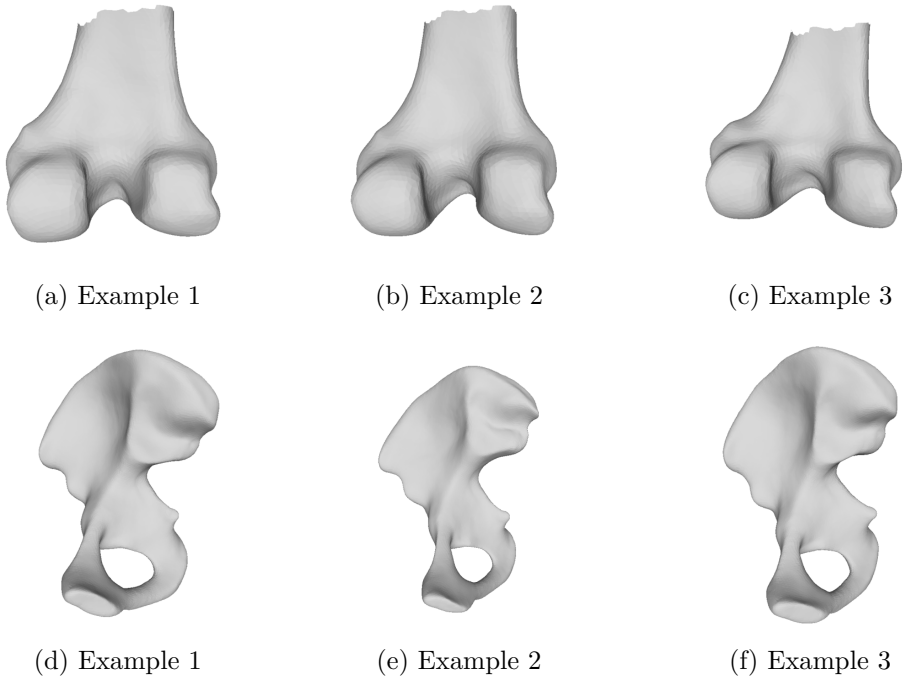
Figure 6.4: Deformations of the pelvis of N7 (R) and N8 (L) and their ground truth. The deformations are also coloured based on their error to the ground truth.

A specific model should only generate shapes that are within the population. To measure the specificity, the idea from Davies is followed, providing a specificity procedure [66]. For each generated shape by a model, the CD to the closest training shape is calculated. The average CD of a model M is said to be a measure of specificity S. For 1000 generated shapes, this can be calculated by:

$$S(M) = \frac{1}{1000} \sum_{j}^{1000} C(x_j, x'_j) \tag{6.1}$$

where x_j depicts a generated shape and x'_j the closest training shape. A model $\mathcal{A}$ is said to be more specific than $\mathcal{B}$ when $S(M_\mathcal{A}) < S(M_\mathcal{B})$. While the goal of the augmentation method is to generate shapes of all kinds of orientations, to measure the specificity the training and generated shapes are first aligned to the template. This way, the measurement of specificity gives a better idea about the generated shapes themselves and is not influenced by the orientation. On the other hand, this also means that the measurement of specificity does not include information about the orientation. It is assumed that the generated orientations are within the population statistics as no visual exorbitancy is seen for any of the cases.

Results of the specificity measurement for different latent embeddings are very similar, but the latent embedding size of 64 gives the most specific model. A value of 0.287 ± 0.04 is obtained. Combined with the generalisability knowledge, a latent embedding of 64 seems to be the best for



(a) Example 1 (b) Example 2 (c) Example 3

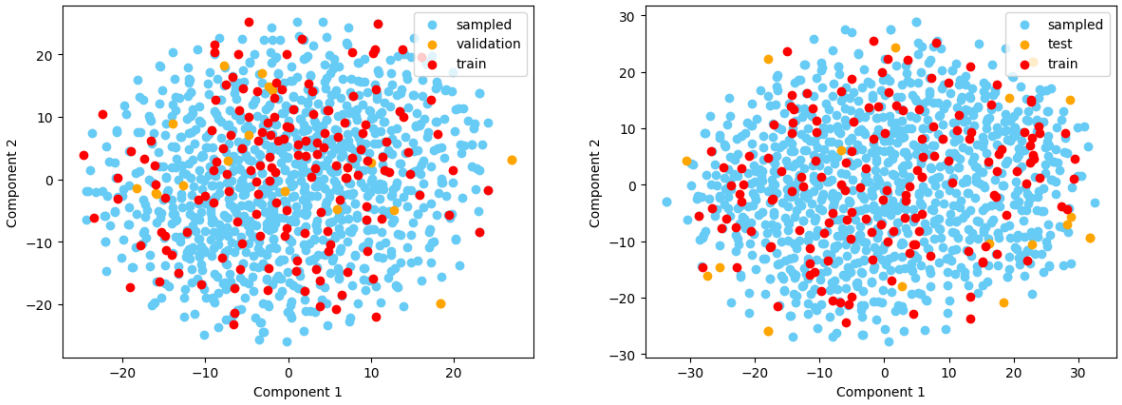
(d) Example 1 (e) Example 2 (f) Example 3

Figure 6.5: Examples of generated shapes of the femur and pelvis using FlowSSM.

further work. However, increasing the latent embedding size brings added complexity. Despite this, the benefits of better generalisability seem to outweigh the added complexity, making it the optimal choice for future work. From now on, a latent embedding size of 64 is used. Next, it is not believed that the model will benefit significantly from a further increase in size.

As the shapes are aligned for this measurement, results can be compared to the shapes generated by a PCA procedure like DeepSSM. 1000 shapes are generated using the DeepSSM framework, fitting a KDE distribution to the PCA space and keeping maximum variability by including all main modes. Bhalodia et al. claim that using a KDE distribution generates plausible shapes while still capturing all the subtleties of the original shape space [48]. Thus it is a good reference to compare the specificity to. The obtained value for these shapes is 0.224 ± 0.01. This shows that a model generating shapes from the PCA space is a bit more specific than the proposed model. Nevertheless, as the found values are within the same order, it can be concluded that the model is able to generate shapes that lie within the population statistics.

For the pelvis, a specificity measurement of 0.274 ± 0.20 is obtained, compared to 0.130 ± 0.01 for the method using PCA. Hence, the PCA method seems to be more specific for more difficult cases like the pelvis. Nevertheless, the generated shapes of the pelvis do visually look acceptable. Examples of generated shapes of both the distal femur and pelvis are shown in Fig. 6.5.



(a) Distal femur (b) Pelvis

Figure 6.6: t-SNE applied on global latent embeddings of both the distal femur and the pelvis.

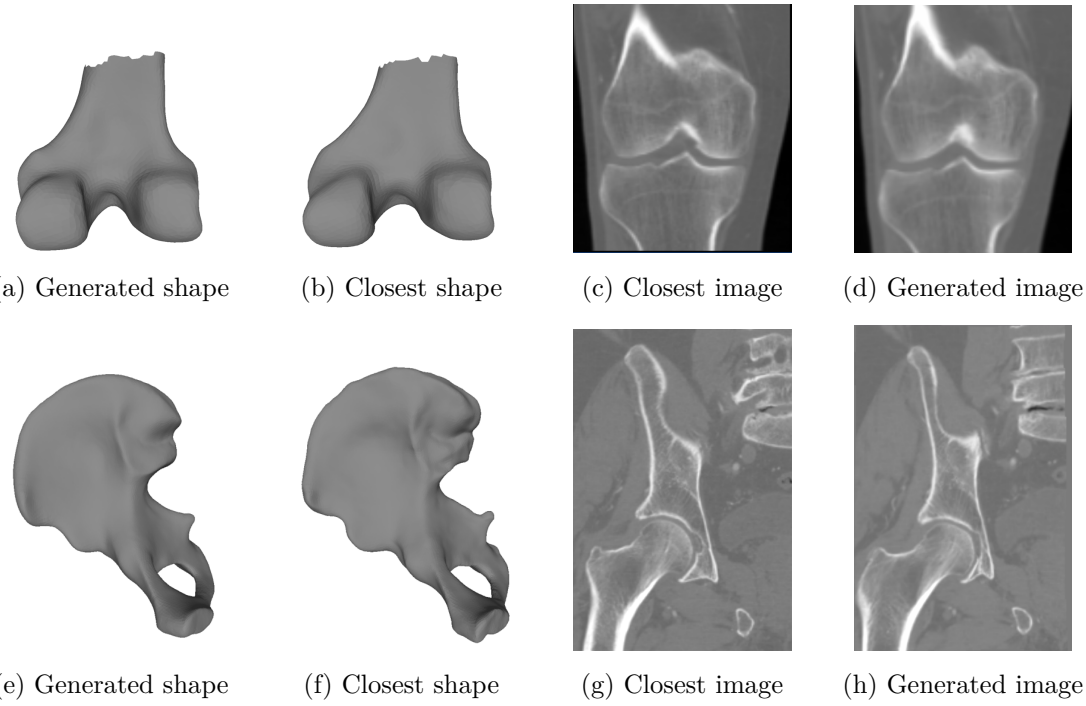
Something that could explain the difference in specificity is that the generated shapes of the pelvis look quite smooth. While the high-frequency deformations were nicely reconstructed for unseen cases, for the generated shapes they seem to be more difficult to obtain. Some extra tuning of the local deformer might be needed to improve this.

The sampling of new shapes happens separately in the global and local latent spaces. To confirm that these spaces are constructed so that they can correctly model the population statistics, the spaces are visualised. The global and local embeddings correspond to shapes of sizes 64 and (64,7,7,7), respectively. To create visualisations, a t-distributed stochastic neighbour embedding (t-SNE), a statistical method for visualising high-dimensional data, is applied on the embeddings. Fig. 6.6 illustrates the global latent spaces after t-SNE for both the distal femur and pelvis, displaying the train, validation, and sampled embeddings [67]. It can be concluded from this analysis that the sampled latent embeddings accurately represent the population statistics. However, some validation cases show latent embeddings at or beyond the borders of the sampled shapes, suggesting that additional flexibility in the sampling method could result in a wider variety of generated shapes.

It can be concluded that FlowSSM is able to generalise well and can generate population-specific shapes, even when shapes are not aligned. This makes it a suitable method for the novel data augmentation technique.

6.2 Generated images

To generate samples comprising both a shape and an image, the shapes generated by FlowSSM are used to warp the original CT images so that they conform to the generated shapes. To deter-



(a) Generated shape (b) Closest shape (c) Closest image (d) Generated image

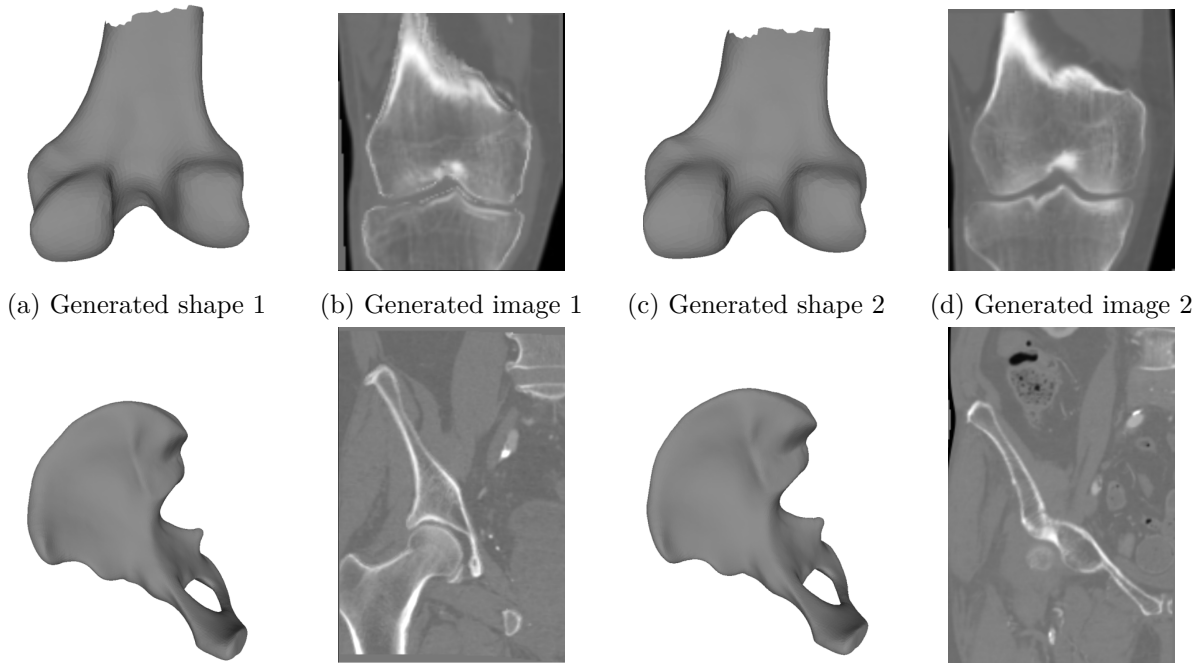
(e) Generated shape (f) Closest shape (g) Closest image (h) Generated image

Figure 6.7: Examples of generated shapes, the closest shape found in the training set, its image, and the final generated image.

mine the closest training shape for each generated shape, the average point-to-point Euclidean distance to all training shapes is calculated and the one with the lowest distance is selected. This enables us to find the TSP warp from the generated shape to the closest training shape.

However, finding the TSP warp requires solving a linear system, of which the number of equations is determined by the number of control points. Therefore, when the number of control points is too large, the TSP warp calculation can become computationally intensive. Consequently, it is important to select the number of control points carefully, as including all vertices as control points can lead to significantly longer processing times. However, using more control points typically results in a more accurate warp. For the distal femur, 2000 random vertices are sampled of the 6151 to find the TSP. This showed to be effective enough to create a good match between the generated shape and warped CT image. As the pelvis shapes have 25967 vertices, a lot more vertices should be sampled to obtain an accurate TSP warp. Due to the fact that using a larger number of vertices would require an unreasonable amount of time, a vertex count of 4800 was chosen for the pelvis. Ideally, a faster method for finding the warp based on all vertices could be used.

Fig. 6.7 shows two examples of a generated shape, the closest training shape, the original training image, and the warped image. Note that it is confirmed that the generated images nicely match the generated shapes.



(a) Generated shape 1 (b) Generated image 1 (c) Generated shape 2 (d) Generated image 2

(e) Generated shape 1 (f) Generated image 1 (g) Generated shape 2 (h) Generated image 2

Figure 6.8: Examples of generated shape and image pairs for both the distal femur and the pelvis.

To conclude this section, some additional examples of the generated shapes and images are showcased in Fig. 6.8. It is worth noting that the image-warping process sometimes results in grey-coloured voxels at the edges with HU value of 0. The aim of image warping is to produce an image output size that matches the original image size. When the original image is scaled down to fit the generated shape, the borders of the image are padded with 0's to achieve the desired size. This can be observed in Figs. 6.8b and 6.8f. While these grey voxels are an undesired effect of the warping, they can be easily eliminated by warping an image larger than the desired size and then cropping it to fit. Nonetheless, the novel approach has yielded excellent shape and image samples.

6.3 Discussion

In this chapter, the results of a novel neural-flow data augmentation method that combines shape generation using FlowSSM with a TSP warp is presented. The power of FlowSSM is leveraged to generate new shapes that follow population statistics. One of the significant advantages of FlowSSM over DeepSSM, which uses PCA, is that there is no constraint to working with aligned shapes. Prior to this work, it was unclear whether FlowSSM could effectively function

on unaligned shapes, as all previous work had included alignment in the data preprocessing stage. It has been demonstrated that alignment is not necessary by applying FlowSSM to two use cases: the distal femur and the pelvis. Mean CDs of 0.177 ± 0.03mm and 0.447 ± 0.04mm were obtained, respectively, demonstrating good generalisation for both simple and complex cases. Notably, for the femur, a lower CD than the authors of FlowSSM was achieved, who obtained a CD of 0.23 ± 0.04mm using aligned shapes. Of course, it is acknowledged that datasets may differ, so direct comparisons should be made with caution. Nonetheless, the results clearly confirm the generalisability of FlowSSM on unaligned shapes.

While generalisability is very important, it is equally essential for the data augmentation application that the models are specific, only generating shapes within the modelled population. In this particular application, utilising a technique such as FlowSSM offers an advantage over PCA methods due to the optimisation of the latent space. The specificity of the method for the distal femur has been evaluated and found to be quite close to the specificity measurement of the DeepSSM PCA method, with a value of 0.287 ± 0.04mm. For the pelvis, however, a specificity measurement of only 0.274 ± 0.2mm was obtained, compared to 0.130 ± 0.01mm for DeepSSM. This may be due to the generated pelvis shapes being smoother than the training shapes. Thus, for more complex cases like the pelvis, further improvements can be made, particularly with the local deformer. However, the model was not optimised for higher specificity, allowing room for future improvements. Although the specificity measurement of the FlowSSM method may be lower than that of the PCA technique used in DeepSSM, a diverse range of shapes with varying sizes and orientations can still be generated, demonstrating the effectiveness of the approach for data augmentation.

It is true that training the network and tuning the hyperparameters requires a significant amount of time, whereas using PCA can be performed almost instantaneously. However, the advantages of the FlowSSM approach go beyond generating shapes with different orientations. It also allows for nonlinearities, which can be highly beneficial for certain applications. Another advantage of FlowSSM over PCA is that it does not require correspondences between shapes. Although the shapes in this case are in correspondence, this is not a requirement for using FlowSSM.

Applying the TSP warps on the images is the next step after shape generation. Choosing the number of vertices for the warp is a challenging task since a larger number of vertices results in higher warp accuracy but also increases the computation time. Therefore, it is important to select the number of vertices carefully based on the shape being used. Additionally, the TSP warp requires correspondences between the shapes, which means that the full potential of using FlowSSM cannot be exploited in this data augmentation method.

Despite the potential for further improvements in specificity and warp accuracy, the neural-flow data augmentation approach has demonstrated great potential. It enables the generation of new training samples for deep learning models that involve both shape and image data.

7 Conclusion and future work

The field of medical image segmentation has seen a significant shift towards the use of deep learning techniques, owing to their ability to deliver quick predictions with high levels of accuracy. The lower limb region is no exception, with automatic bone segmentation methods gaining popularity. Despite their potential, achieving a balance between speed and accuracy remains a challenging task. While some studies focus on specific ROIs, the need for manual cropping renders them less clinically practical. Conversely, studies that employ whole-body CT scans often compromise on either accuracy or speed, making them less reliable for clinical applications.

In this thesis, a novel, fully automatic workflow that addresses the challenge of achieving both rapid and precise bone segmentation in a whole-body CT scan of the lower limb, encompassing the pelvis, femur, tibia, and fibula was presented. The proposed workflow achieved high levels of accuracy by first localising the regions of interest based on a downsampled image. Subsequently, the ROIs are segmented in full resolution. The workflow incorporated a total of four models: one for downsampled predictions and three for more detailed segmentation.

To assess the efficacy of the method, the workflow was developed and validated using an in-house dataset comprising high-quality CT images of healthy patients. The majority of the images were obtained from the same hospital and scanner type, with a few exceptions included. Our validation process demonstrated that the proposed workflow delivers faster predictions than existing state-of-the-art methods, with comparable levels of accuracy.

Despite the promising results of our workflow, certain issues have been identified during its implementation. Notably, in some cases, leakage of the femur in the knee joint region and the pelvis in the pubic symphysis has been observed. Additionally, our employed models tend to predict some voxels as bone that are far from the ground truth. Although these discrepancies could be attributed to label maps that are not always accurate or insufficient optimisation, they may also be inherent to the models themselves. These issues also lead to some of the possible adaptations and improvements that can be made in further research. To obtain a clinically

relevant tool, these improvements are necessary:

- Although bone localisation has been consistently effective across all test cases, there is still considerable scope for improvement in the full-resolution segmentations. To enhance the quality of the full-resolution segmentations, it is recommended to refine the preprocessing workflow to account for possible bounding box errors in the localisation step and incorporate greater variability in the images used for training bone-specific models. Additionally, optimising the network architecture and training procedure for each bone separately could be explored. For instance, a combination of the cross-entropy and dice loss functions could improve performance. Moreover, a deeper architecture with more parameters and a patch-based training approach to mitigate memory constraints could also contribute to better segmentation results.

- While the bone-specific models can be improved, it is still possible that they may misclassify some voxels or encounter issues such as leakage. This is because these models lack prior anatomical information. To address this, it may be beneficial to apply a SSM fitting to the deep learning predictions as a postprocessing step, similar to the approach taken by Ambellan et al. [8]. By fitting an SSM, it is possible to remove misclassified voxels and correct leakage. However, this could lead to longer prediction times.

- The workflow has been developed to include the major bones in the lower limb region. In future work, additional bones such as the sacrum or patella could be incorporated to create a comprehensive segmentation tool that includes all the bones in the lower limb region. However, the current downsampled method may not be suitable for smaller bones as they may not be clearly visible on downsampled images. An alternative approach to finding bounding boxes for these bones could be to use anatomical knowledge to create the boxes based on the current bounding boxes, rather than relying on the downsampling method. This would help to ensure that all bones in the lower limb region are accurately segmented.

- While the workflow demonstrated promising results, it is essential to validate it on different datasets to ensure its clinical relevance. Validation on a diverse range of datasets can help to establish the generalisability of the workflow and ensure that it can be applied effectively in clinical settings.

In addition to proposing a segmentation workflow, a novel data augmentation method was also introduced. The primary goal of this method was to enhance the segmentation process by addressing the potential limitations associated with a lack of data. However, the adaptations above seem to be more effective as current results are better than expected. Nevertheless, the data augmentation method was further explained as certain applications might still benefit

from it. For example, an application where shapes and latent representations, could directly be predicted from images like DeepSSM is trying could benefit from this method.

In contrast to the DeepSSM data augmentation method that employs PCA for shape generation, the novel data augmentation technique utilises FlowSSM for shape generation. FlowSSM is a neural flow method that leverages deformers to learn to deform a template shape to a target shape while optimising the latent embedding. This enables the sampling of new latent embeddings from the created latent spaces. By passing these embeddings through the trained deformers, new shapes that adhere to the population statistics can be generated. To complete the data augmentation process, a TSP warp is computed from the generated shape to the nearest training shape and applied to the corresponding training image, creating new shape and image pairs.

The use of FlowSSM for shape generation has proven to be highly effective, with the added benefit of eliminating the need for shape alignment, unlike the use of PCA. Moreover, FlowSSM has the ability to introduce nonlinear shape generation, unlike PCA which can only capture linearities. The application of FlowSSM to the distal femur and the pelvis has shown promising results in terms of both generalisation and specificity. However, one challenge encountered during the use of FlowSSM is that the generated shapes of the pelvis appear smoother than the training shapes, likely due to the lack of optimisation of the training procedure. One of the difficulties of using FlowSSM is that a large number of hyperparameters need to be tuned, but a tuning procedure proposed by the authors is available to address this issue. Despite being a challenging case compared to the distal femur, the pelvis has shown promising outcomes, indicating the potential for future applications.

While another benefit of using FlowSSM is that no correspondence o the shapes is needed, applying a TSP warp requires correspondence. The detail of the warp relies on the number of control points. A larger number gives more detailed warps but also requires a larger calculation time.

Even though this method proves to be successful for even a difficult case like the pelvis, some improvements or adaptions are possible:

- Improved standardisation of the large-scale hyperparameter tuning process for FlowSSM could enhance the effectiveness of the augmentation method, since different cases may require different hyperparameters. Consequently, using this method for multiple cases simultaneously would benefit from a more streamlined tuning process.
- The utilisation of a TSP warp negates the advantage of FlowSSM, which eliminates the need for correspondence. Therefore, discovering a warp between the shapes that does not require this correspondence could result in an even more advantageous data augmentation

method.

- To obtain the sampled latent embeddings, one approach is to perform a PCA on the latent space and use a linear combination of the main modes with randomly initialised weights. However, an alternative and potentially more effective approach is to learn the distribution of the latent space and sample from it. This enables models to become more specific. For instance, mixture density networks can be utilised to learn the distribution and associated uncertainties, as discussed by Brando [68].
- Although this thesis focused on cases with a single shape, CT images typically contain multiple anatomically relevant structures. Therefore, it would be intriguing to explore scenarios with multiple objects or cases where a single object is comprised of disconnected regions. It is currently unclear how FlowSSM would handle these situations.

Both the segmentation and data augmentation techniques demonstrated promising results, suggesting the potential for developing tools that could be clinically significant in the future. Nevertheless, to fully realise this potential, additional improvements or adaptations must be implemented to further refine the methods and enhance their accuracy and reliability.

Bibliography

[1] E. A. Audenaert, J. Van Houcke, D. F. Almeida, L. Paelinck, M. Peiffer, G. Steenackers, and D. Vandermeulen, "Cascaded statistical shape model based segmentation of the full lower limb in CT," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 22, no. 6, pp. 644–657, 2019.

[2] C. R. Henak, A. E. Anderson, and J. A. Weiss, "Subject-specific analysis of joint contact mechanics: Application to the study of osteoarthritis and surgical planning," *Journal of Biomechanical Engineering*, vol. 135, no. 2, pp. 1–26, 2013.

[3] M. Hans-Peter and T. Heimann, "Statistical shape models for 3d medical image segmentation: {A} review," *Medical Image Analysis*, vol. 13, pp. 543–563, 2009.

[4] C. Chu, C. Chen, L. Liu, and G. Zheng, "FACTS: Fully Automatic CT Segmentation of a Hip Joint," *Annals of Biomedical Engineering*, vol. 43, no. 5, pp. 1247–1259, 2015.

[5] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, no. December 2012, pp. 60–88, 2017.

[6] C. D. Naylor, "On the prospects for a (Deep) learning health care system," *JAMA - Journal of the American Medical Association*, vol. 320, no. 11, pp. 1099–1100, 2018.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015.

[8] F. Ambellan, A. Tack, M. Ehlke, and S. Zachow, "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative," *Medical Image Analysis*, vol. 52, pp. 109–118, 2019.

[9] P. Liu, H. Han, Y. Du, H. Zhu, Y. Li, F. Gu, H. Xiao, J. Li, C. Zhao, L. Xiao, X. Wu, and S. K. Zhou, "Deep learning to segment pelvic bones: large-scale CT datasets and baseline models," *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, no. 5, pp. 749–756, 2021.

[10] 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 and Biological Engineering and Computing*, vol. 60, no. 5, pp. 1417–1429, 2022.

[11] 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, no. 1, pp. 21–29, 2019.

[12] R. J. Kuiper, R. J. Sakkers, M. van Stralen, V. Arbabi, M. A. Viergever, H. Weinans, and P. R. Seevinck, "Efficient cascaded v-net optimization for lower extremity ct segmentation validated using bone morphology assessment," *Journal of Orthopaedic Research*, vol. 40, pp. 2894–2907, 12 Dec. 2022.

[13] E. Marieb and K. Hoehn, *Human Anatomy & Physiology*, 11th ed. Pearson Eduction, 2019, ch. The Skeleton, pp. 231–282.

[14] J. M. DeSilva and K. R. Rosenberg, "Anatomy, development, and function of the human pelvis," *The Anatomical Record*, vol. 300, no. 4, pp. 628–632, 2017.

[15] F. Netter, *Netter Atlas of Human Anatomy: Classic Regional Approach*, 8th ed. Elsevier - Health Sciences Division, 2022.

[16] T. Fukuda, T. Yonenaga, T. Miyasaka, T. Kimura, M. Jinzaki, and H. Ojiri, "CT in osteoarthritis: its clinical role and recent advances," *Skeletal Radiology*, no. 0123456789, 2022.

[17] A. Virzì, C. O. Muller, J. B. Marret, E. Mille, L. Berteloot, D. Grévent, N. Boddaert, P. Gori, S. Sarnacki, and I. Bloch, "Comprehensive review of 3d segmentation software tools for mri usable for pelvic surgery planning," *Journal of Digital Imaging*, vol. 33, pp. 99–110, 1 Feb. 2020.

[18] N. J. Mardis, "Emerging technology and applications of 3D printing in the medical field," en, *Missouri medicine*, vol. 115, no. 4, pp. 368–373, Jul. 2018.

[19] L. Pugliese, S. Marconi, E. Negrello, V. Mauri, A. Peri, V. Gallo, F. Auricchio, and A. Pietrabissa, "The clinical use of 3D printing in surgery," *Updates in Surgery*, vol. 70, no. 3, pp. 381–388, 2018.

[20] P. De Backer, S. Vermijs, C. Van Praet, P. De Visschere, S. Vandenbulcke, A. Mottaran, C. A. Bravi, C. Berquin, E. Lambert, S. Dautricourt, W. Goedertier, A. Mottrie, C. Debbaut, and K. Decaestecker, "A Novel Three-dimensional Planning Tool for Selective Clamping During Partial Nephrectomy: Validation of a Perfusion Zone Algorithm," *European Urology*, vol. 83, pp. 413–421, 2023.

[21] H. Lamecker, "Variational and Statistical Shape Modeling for 3D Geometry Reconstruction," *Naturwissenschaften*, p. 123, 2008.

[22] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[23] M. Gao, N. Ruan, J. Shi, and W. Zhou, "Deep Neural Network for 3D Shape Classification Based on Mesh Feature," *Sensors*, vol. 22, no. 18, pp. 1–11, 2022.

[24] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Eurographics Symposium on Geometry Processing*, vol. 30, no. 6, pp. 1681–1707, 2011.

[25] Y. Sahillioğlu, "Recent advances in shape correspondence," *Visual Computer*, vol. 36, no. 8, pp. 1705–1721, 2020.

[26] T. Heimann and H. P. Meinzer, "Statistical shape models for 3D medical image segmentation: A review," *Medical Image Analysis*, vol. 13, no. 4, pp. 543–563, 2009.

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

[28] T. Panch, P. Szolovits, and R. Atun, "Artificial intelligence, machine learning and health systems," *Journal of Global Health*, vol. 8, no. 2, pp. 1–8, 2018.

[29] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," *International Conference on ICT and Knowledge Engineering*, pp. 1–6, 2018.

[30] H. D. Block, "The perceptron: A model for brain functioning," *Reviews of Modern Physics*, vol. 34, pp. 123–135, 1962.

[31] D. Plaut, S. Nowlan, and G. Hinton, "Experiments on learning by back propagation," *Technical Report CMU-CS-86-126*, no. June, 1986.

[32] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016.

[33] J. L. Ba and D. P. Kingma, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.

[34] R. M. Schmidt, F. Schneider, and P. Hennig, "Descending through a crowded valley - benchmarking deep learning optimizers," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 9367–9376.

[35] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015.

[36] M. A. Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. D. B. Bruce, "Position, padding and predictions: A deeper look at position information in cnns," *CoRR*, vol. abs/2101.12322, 2021.

[37] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," *CoRR*, vol. abs/2009.07485, 2020.

[38] D. L. Phamé, C. Xué, and J. L. Princeé, "A survey of current methods in medical image segmentation," 1998.

[39] L. Lenchik, L. Heacock, A. A. Weaver, R. D. Boutin, T. S. Cook, J. Itri, C. G. Filippi, R. P. Gullapalli, J. Lee, M. Zagurovskaya, T. Retson, K. Godwin, J. Nicholson, and P. A. Narayana, "Automated segmentation of tissues using ct and mri: A systematic review," *Academic Radiology*, vol. 26, pp. 1695–1706, 12 Dec. 2019.

[40] D. Marzorati, M. Sarti, L. Mainardi, A. Manzotti, and P. Cerveri, "Deep 3D convolutional networks to segment bones affected by severe osteoarthritis in CT scans for PSI-based knee surgical planning," *IEEE Access*, vol. 8, pp. 196 394–196 407, 2020.

[41] M. Krčah, G. Székely, and R. Blanc, "Fully automatic and fast segmentation of the femur bone from 3D-CT images with no shape prior," *Proceedings - International Symposium on Biomedical Imaging*, pp. 2087–2090, 2011.

[42] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, no. 2000, pp. 315–337, 2000.

[43] N. Otsu, P. L. Smith, D. B. Reid, C. Environment, L. Palo, P. Alto, and P. L. Smith, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. C, no. 1, pp. 62–66, 1979.

[44] J. Zhang, C. H. Yan, C. K. Chui, and S. H. Ong, "Fast segmentation of bone in CT images using 3D adaptive thresholding," *Computers in Biology and Medicine*, vol. 40, no. 2, pp. 231–236, 2010.

[45] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *Journal of the Royal Statistical Society*, vol. 53, no. 2, pp. 285–339, 1991.

[46] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, 2021.

[47] J. Cates, S. Elhabian, and R. Whitaker, "Shapeworks: Particle-based shape correspondence and visualization software," in *Statistical Shape and Deformation Analysis*, Elsevier, 2017, pp. 257–298.

[48] R. Bhalodia, S. Elhabian, J. Adams, W. Tao, L. Kavan, and R. Whitaker, "DeepSSM: A Blueprint for Image-to-Shape Deep Learning Models," 2021.

[49] A. Adam, *Mesh voxelisation*, `https://nl.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation`, 2023.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[51] *BRATS challenge*, `https://www.med.upenn.edu/sbia/brats2018.html`, 2018.

[52] *ISLES challenge*, `http://www.isles-challenge.org/ISLES2018/`, 2019.

[53] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric Analysis of White Matter Lesions in MR Images: Method and Validation," *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.

[54] D. P. Huttenlocher, W. J. Rucklidge, and G. A. Klanderman, "Comparing images using the Hausdorff distance under translation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1992-June, no. 9, pp. 654–656, 1992.

[55] W. E. Lorensen and H. E. Cline, "Marching Cubes: a High Resolution 3D Surface Construction Algorithm.," *Computer Graphics (ACM)*, vol. 21, no. 4, pp. 163–169, 1987.

[56] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "Scikit-image: Image processing in python," *PeerJ*, vol. 2, e453, 2014.

[57] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds., The Eurographics Association, 2008.

[58] G. Taubin, "Signal processing approach to fair surface design," *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 351–358, 1995.

[59] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 5932–5941, 2019.

[60] F. Chen, Y. Xie, P. Xu, Z. Zhao, D. Zhang, and H. Liao, "Efficient lower-limb segmentation for large-scale volumetric ct by using projection view and voxel group attention," *Medical and Biological Engineering and Computing*, vol. 60, pp. 2201–2216, 8 Aug. 2022.

[61] D. Lüdke, T. Amiranashvili, F. Ambellan, I. Ezhov, B. H. Menze, and S. Zachow, "Landmark-free statistical shape modeling via neural flow deformations," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, Springer Nature Switzerland, 2022, pp. 453–463.

[62] F. L. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[63] H. Huy Khahn, *Thin Plate Splines Warping*, `https://khanhha.github.io/posts/Thin-Plate-Splines-Warping`, 2019.

[64] D. C. Lüdke, "Neural flow-based deformations for statistical shape modelling," 2022, Thesis FLOW SSM.

[65] K. Gupta and M. Chandraker, "Neural mesh flow: 3D manifold mesh generation via diffeomorphic flows," *Advances in Neural Information Processing Systems*, vol. 2020-December, no. NeurIPS, pp. 1–12, 2020.

[66] R. Davies, "Learning shape: optimal models for analysing natural variability," *Learning*, 2002.

[67] L. V. D. Maaten and G. Hinton, "Visualizing Data using t-SNE," vol. 9, pp. 2579–2605, 2008.

[68] A. Brando, *Mixture density networks (mdn) for distribution and uncertainty estimation*, Report of the Master's Thesis: Mixture Density Networks for distribution and uncertainty estimation., 2017.

[69] V. Jorgetti, L. M. Dos Reis, and S. M. Ott, "Ethnic differences in bone and mineral metabolism in healthy people and patients with CKD," *Kidney International*, vol. 85, no. 6, pp. 1283–1289, 2014.

A Appendix

A.1 Ethical reflection

This thesis places significant emphasis on ethical considerations, recognising their vital importance, particularly because this thesis aims to develop a tool that could potentially have a direct impact on individuals in a clinical setting in the future. In this appendix, not only are the aspects concerning the utilised data discussed but also a thoughtful analysis is conducted regarding the potential impact of the obtained results and the maintenance of scientific integrity.

A.1.1 Ethical aspects directly related to the research

In Chapter 3, we focused on the dataset provided by Prof. Audenaert from the UH Ghent, which comprises both whole-body CT scans and 3D models of the lower limb bones for 94 subjects. The CT scans were collected from various hospitals, with a primary source being AZ Groeninge in Kortrijk, Belgium, between January 2009 and November 2020. It is important to note that the scans are stored in DICOM format, which includes not only imaging data but also personal information. Consequently, the data necessitates careful handling and protection of privacy.

The scans in this study underwent an anonymisation process to ensure privacy. Names were removed, and each scan was assigned a random subject ID. The only personal information available in the metadata is the patient's age and sex. As a result, it is not possible for data users to establish a direct link between the images and specific individuals, thereby allowing for further utilisation of the data. The ages of the subjects range from 39 to 96 years, with an average age of 66.7 years. It is worth noting that there is a slight inclination towards older participants. The dataset consists of 53 males, 40 females, and 1 nonbinary individual, with males forming a slight majority. Given that bone differences, particularly in the pelvis, can be observed between

males and females, this aspect should be taken into consideration during analysis. However, we believe that since there are enough female subjects included, any potential impact on the results due to differences in bones between males and females should not be significant.

We assume that images accurately represent the Belgian population within the found age range. However, it is essential to recognise that this study primarily focuses on the population found in Belgium, possibly limiting its generalisability to other populations. Therefore, it is crucial to validate the methods used in this study on datasets derived from diverse populations. Differences in bone densities have been observed among various ethnicities, potentially resulting in CT scans that differ from those used in this study [69].

A.1.2 Reflection about the potential impact of results

This thesis aims to develop a segmentation tool that can rapidly and accurately predict the major bones in the lower limb region. The primary objective is to provide real-time subject-specific data, which could be valuable for future clinicians. Such fast and accurate predictions have the potential to benefit clinical practice by informing patients during consultations and aiding in diagnostics, treatment planning, and surgical procedures. Moreover, these results could also contribute to further research investigating the effects of bone anatomy. However, it is important to note that despite the progress made, some errors still exist in the current results. Consequently, it is not yet suitable for clinical use. At this point of the research, this tool will not be used without any rigorous checks and validation. Further improvements and adaptations are necessary to refine the segmentation tool, with the ultimate goal of creating a stand-alone tool that can be truly beneficial in clinical settings.

This research specifically targeted healthy patients, excluding individuals with fractures or diseases such as osteoarthritis. Consequently, the results obtained are only applicable to specific cases involving healthy patients. Therefore, it is not intended for a clinical setting encompassing a diverse range of patients with varying conditions. Nevertheless, the techniques developed in this research have the potential to be adapted and utilised for cases involving patients with different conditions, provided that suitable data is available. There is a strong belief that with further improvements and access to more extensive datasets, these deep learning techniques can effectively handle a wide range of cases, including those with fractures or diseases.

A.1.3 Scientific integrity

Maintaining scientific integrity is of utmost importance in this work. We strive to uphold honesty and transparency in all aspects presented. Whenever utilising images sourced from external origins, proper citations are provided. Similarly, the developed methods rely on specific software

packages or techniques created by various authors, and credit is given through appropriate referencing. If any author feels that a particular reference is missing, it is crucial to highlight this omission, as it is never intentional. We value the input and collaboration of fellow researchers and are open to rectifying any mistakes.

In this work, all available data was utilised unless specific preprocessing errors occurred, which were limited to three cases. These cases are explicitly listed for transparency. Therefore, other researchers working with similar data can easily replicate all steps followed in this study. When reporting the outcomes of the study, all subjects are included to ensure comprehensive and unbiased analysis. Appropriate metrics are employed to demonstrate a minimal bias towards certain majority classes within the data. To prevent overfitting, an independent test set is employed to evaluate the performance of the models, ensuring they are not solely optimised for the training data. However, it is important to note that further validation using new and diverse datasets is a planned future step.

Visualisations are consistently generated to illustrate both the best and worst cases, whenever feasible. This approach ensures that no mistakes or shortcomings are concealed. In instances where poor results are observed, they are thoroughly examined, and the reasons behind their occurrence are explained in detail.

A.2 Dataset information

Patient ID	Hospital	Scanner Manufacturer	Sex	Age	Slice Thickness	Spacing
N1	AZ Groeninge	GE MEDICAL SYSTEMS	M	73	0.625	0.625
N2	AZ Groeninge	GE MEDICAL SYSTEMS	F	62	0.625	0.660
N3	AZ Groeninge	GE MEDICAL SYSTEMS	M	79	0.625	0.719
N4	AZ Groeninge	GE MEDICAL SYSTEMS	M	78	0.625	0.625
N5	AZ Groeninge	GE MEDICAL SYSTEMS	M	70	0.625	0.625
N7	AZ Groeninge	GE MEDICAL SYSTEMS	M	73	0.625	0.625
N8	AZ Groeninge	GE MEDICAL SYSTEMS	M	64	0.625	0.625
N9	AZ Groeninge	GE MEDICAL SYSTEMS	M	51	0.625	0.625
N10	AZ Groeninge	GE MEDICAL SYSTEMS	M	74	0.625	0.766
N11	AZ Groeninge	GE MEDICAL SYSTEMS	F	77	0.625	0.625
N12	AZ Groeninge	GE MEDICAL SYSTEMS	M	78	0.625	0.625
N14	AZ Groeninge	GE MEDICAL SYSTEMS	M	43	0.625	0.678
N15	AZ Groeninge	GE MEDICAL SYSTEMS	F	87	0.625	0.719
N16	AZ Groeninge	GE MEDICAL SYSTEMS	M	80	0.625	0.672
N17	AZ Groeninge	GE MEDICAL SYSTEMS	M	54	0.625	0.625
N18	AZ Groeninge	GE MEDICAL SYSTEMS	M	52	0.625	0.672
N19	AZ Groeninge	GE MEDICAL SYSTEMS	F	82	0.625	0.654

N20	AZ Groeninge	GE MEDICAL SYSTEMS	F	70	0.625	0.660
N21	AZ Groeninge	GE MEDICAL SYSTEMS	M	50	0.625	0.625
N23	AZ Groeninge	GE MEDICAL SYSTEMS	F	00	0.625	0.625
N24	AZ Groeninge	GE MEDICAL SYSTEMS	M	71	0.625	0.625
N25	AZ Groeninge	GE MEDICAL SYSTEMS	M	60	0.625	0.840
N26	AZ Groeninge	GE MEDICAL SYSTEMS	M	52	0.625	0.723
N27	AZ Groeninge	GE MEDICAL SYSTEMS	M	67	0.625	0.625
N28	AZ Groeninge	GE MEDICAL SYSTEMS	M	50	0.625	0.625
N29	AZ Groeninge	GE MEDICAL SYSTEMS	M	66	0.625	0.625
N30	AZ Groeninge	GE MEDICAL SYSTEMS	M	54	0.625	0.766
N31	AZ Groeninge	GE MEDICAL SYSTEMS	M	57	0.625	0.625
N33	AZ Groeninge	GE MEDICAL SYSTEMS	M	54	0.625	0.625
N34	AZ Groeninge	GE MEDICAL SYSTEMS	M	74	0.625	0.672
N35	AZ Groeninge	GE MEDICAL SYSTEMS	F	39	0.625	0.625
N36	AZ Groeninge	GE MEDICAL SYSTEMS	F	74	0.625	0.625
N37	AZ Groeninge	GE MEDICAL SYSTEMS	F	72	0.625	0.742
N38	AZ Groeninge	GE MEDICAL SYSTEMS	F	60	0.625	0.625
N39	AZ Groeninge	GE MEDICAL SYSTEMS	F	86	0.625	0.760
N40	AZ Groeninge	GE MEDICAL SYSTEMS	F	75	0.625	0.643

N41	AZ Groeninge	GE MEDICAL SYSTEMS	F	70	0.625	0.625
N42	AZ Groeninge	GE MEDICAL SYSTEMS	M	80	0.625	0.625
N43	AZ Groeninge	GE MEDICAL SYSTEMS	M	54	0.625	0.672
N44	AZ Groeninge	GE MEDICAL SYSTEMS	F	87	0.625	0.625
N45	AZ Groeninge	GE MEDICAL SYSTEMS	M	96	0.625	0.812
N46	AZ Groeninge	GE MEDICAL SYSTEMS	M	79	0.625	0.742
N47	AZ Groeninge	GE MEDICAL SYSTEMS	F	77	0.625	0.625
N49	AZ Groeninge	GE MEDICAL SYSTEMS	F	68	0.625	0.578
N50	AZ Groeninge	GE MEDICAL SYSTEMS	M	49	0.625	0.625
N51	AZ Groeninge	GE MEDICAL SYSTEMS	M	83	0.625	0.678
N52	AZ Groeninge	GE MEDICAL SYSTEMS	F	49	0.625	0.625
N53	AZ Groeninge	GE MEDICAL SYSTEMS	F	81	0.625	0.730
N54	AZ Groeninge	GE MEDICAL SYSTEMS	F	80	0.625	0.818
N56	AZ Groeninge	GE MEDICAL SYSTEMS	M	62	0.625	0.625
N57	AZ Groeninge	GE MEDICAL SYSTEMS	F	91	0.625	0.625
N58	AZ Groeninge	GE MEDICAL SYSTEMS	M	74	0.625	0.760
N60	AZ Groeninge	GE MEDICAL SYSTEMS	M	54	0.625	0.695
N61	AZ Groeninge	GE MEDICAL SYSTEMS	M	81	0.625	0.660
N62	AZ Groeninge	GE MEDICAL SYSTEMS	F	72	0.625	0.625

N64	AZ GROENINGE	GE MEDICAL SYSTEMS	M	60	1.250	0.639
N65	AZ Groeninge	GE MEDICAL SYSTEMS	F	78	0.625	0.625
N66	AZ Groeninge	GE MEDICAL SYSTEMS	M	46	0.625	0.625
N69	AZ Groeninge	GE MEDICAL SYSTEMS	F	79	0.625	0.730
N70	AZ Groeninge	GE MEDICAL SYSTEMS	M	88	0.625	0.625
N71	Unkown	Unkown	M	82	1.250	0.703
N72	Unkown	Unkown	F	44	1.250	0.834
N73	Unkown	Unkown	M	60	1.250	0.756
N74	Unkown	Unkown	M	63	1.250	0.662
N75	UZ Gent	SIEMENS	M	57	1.500	0.740
N76	Unkown	Unkown	F	48	1.250	0.971
N78	Unkown	Unkown	M	63	1.250	0.750
N81	Unkown	Unkown	F	64	1.250	0.703
N82	Unkown	Unkown	M	78	1.250	0.703
N84	Unkown	Unkown	F	57	1.250	0.719
N85	Unkown	Unkown	F	58	1.250	0.775
N86	Unkown	GE MEDICAL SYSTEMS	F	48	1.250	0.789
N87	Unkown	GE MEDICAL SYSTEMS	F	80	1.250	0.703
N88	Unkown	GE MEDICAL SYSTEMS	M	60	1.250	0.764
N89	Unkown	GE MEDICAL SYSTEMS	M	74	1.250	0.947
N91	Unkown	GE MEDICAL SYSTEMSS	M	65	1.250	0.859
N92	Unkown	GE MEDICAL SYSTEMS	M	61	1.250	0.906
N93	Unkown	GE MEDICAL SYSTEMS	F	73	1.250	0.881
N94	Unkown	GE MEDICAL SYSTEMS	F	59	1.250	0.760

N95	Unkown	GE MEDICAL SYSTEMS	M	86	1.250	0.703
N96	Unkown	GE MEDICAL SYSTEMS	F	71	1.250	0.703
N97	Unkown	GE MEDICAL SYSTEMS	F	80	1.250	0.703
N98	Unkown	GE MEDICAL SYSTEMS	M	69	1.250	0.873
N100	Unkown	GE MEDICAL SYSTEMS	M	78	1.250	0.703
N101	Unkown	GE MEDICAL SYSTEMS	F	47	1.250	0.703
N102	Unkown	GE MEDICAL SYSTEMS	F	63	1.250	0.703
N111	AZ Glorieux	GE MEDICAL SYSTEMS	O	68	0.625	0.686
N113	St Andries Tielt	GE MEDICAL SYSTEMS	F	71	0.625	0.703
N114	St Andries Tielt	GE MEDICAL SYSTEMS	M	74	0.625	0.777
N115	St Andries Tielt	GE MEDICAL SYSTEMS	M	50	0.625	0.703
N116	St Andries Tielt	GE MEDICAL SYSTEMS	M	71	0.625	0.773
N117	St Andries Tielt	GE MEDICAL SYSTEMS	F	86	0.625	0.703
N118	St Andries Tielt	GE MEDICAL SYSTEMS	F	67	0.625	0.703
N119	St Andries Tielt	GE MEDICAL SYSTEMS	F	52	0.625	0.605

A.3 Segmentation evaluation metrics

Type	N12	N14	N15	N16	N17	N18	N19	N20	N21
Femur R	0.989	0.982	0.983	0.984	0.985	0.987	0.986	0.986	0.986
Femur L	0.990	0.981	0.981	0.99	0.984	0.988	0.984	0.987	0.980
Pelvis R	0.971	0.973	0.972	0.973	0.980	0.975	0.975	0.972	0.975
Pelvis L	0.98	0.957	0.971	0.973	0.980	0.977	0.975	0.959	0.976
Tibia + Fibula R	0.980	0.978	0.973	0.981	0.978	0.982	0.962	0.971	0.973
Tibia + Fibula L	0.980	0.975	0.972	0.979	0.977	0.979	0.976	0.975	0.978

Table A.2: DSC of all subjects and bone classes.

Type	N12	N14	N15	N16	N17	N18	N19	N20	N21
Femur R	0.989	0.993	0.987	0.986	0.989	0.99	0.983	0.992	0.995
Femur L	0.993	0.985	0.985	0.992	0.987	0.989	0.986	0.988	0.995
Pelvis R	0.976	0.981	0.973	0.974	0.985	0.987	0.983	0.977	0.978
Pelvis L	0.987	0.963	0.971	0.974	0.985	0.992	0.987	0.971	0.982
Tibia + Fibula R	0.981	0.981	0.976	0.981	0.986	0.983	0.965	0.978	0.979
Tibia + Fibula L	0.981	0.981	0.98	0.986	0.982	0.985	0.978	0.975	0.985

Table A.3: Precision of all subjects and bone classes.

Type	N12	N14	N15	N16	N17	N18	N19	N20	N21
Femur R	0.989	0.972	0.978	0.982	0.982	0.983	0.990	0.980	0.978
Femur L	0.986	0.977	0.977	0.989	0.981	0.988	0.982	0.985	0.965
Pelvis R	0.967	0.965	0.972	0.972	0.975	0.963	0.967	0.967	0.971
Pelvis L	0.973	0.95	0.971	0.972	0.974	0.963	0.963	0.947	0.970
Tibia + Fibula R	0.978	0.975	0.970	0.982	0.970	0.980	0.958	0.965	0.966
Tibia + Fibula L	0.979	0.969	0.964	0.972	0.972	0.974	0.973	0.976	0.971

Table A.4: Recall of all subjects and bone classes.

Type	N12	N14	N15	N16	N17	N18	N19	N20	N21
Femur R	4.69	1.732	5.196	0.0	1.732	3.464	1.732	10.392	5.831
Femur L	5.196	6.928	10.392	1.732	5.196	3.464	1.414	3.464	5.196
Pelvis R	6.403	15.588	13.856	13.856	10.392	1.732	3.464	3.0	3.464
Pelvis L	3.0	36.373	90.648	3.464	8.124	21.378	19.053	17.321	1.732
Tibia + Fibula R	6.928	1.732	3.0	17.321	5.196	4.123	1.0	4.123	3.464
Tibia + Fibula L	22.517	3.0	0.0	0.0	3.0	1.732	3.0	1.0	3.464

Table A.5: HD of all subjects and bone classes.

FACULTY OF ENGINEERING AND ARCHITECTURE

Lower Limb Bone Segmentation through Deep Learning and Neural Flow Based Data Augmentation

Roel Huysentruyt
Student number: 01710473

Supervisors: Prof. dr. ir. Aleksandra Pizurica, Prof. dr. Emmanuel Audenaert
Counsellors: Srdan Lazendic, Ide Van den Borre

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Biomedical Engineering

Academic year 2022-2023

GHENT UNIVERSITY

VUB VRIJE UNIVERSITEIT BRUSSEL