Development and Model Form Assessment of an Automatic Subject-specific Vertebra Reconstruction Method

Dingzhong Zhang

Department of Mechanical Engineering

McGill University

Montréal, Québec, Canada

July 26, 2022

A thesis submitted to McGill University in partial fulfilment of the requirements of the degree of Master of Science in Mechanical Engineering

Abstract

Background: Current spine models for analog bench models, surgical navigation and surgical training platforms are conventionally based on 3D models produced from anatomical human body polygon database or reconstructed from manual-labelled subject-specific scanning data. A quick and accurate reconstruction method for subject-specific spine models is important as often such platforms are leveraged to develop or improve treatments. In the meantime, the conventional 3D model evaluation metrics can only reflect overall static model accuracy and lack the specificity to evaluate the model accuracy from different perspectives.

Objective: Propose a workflow of automatic subject-specific vertebra reconstruction method and quantify the reconstructed model accuracy and model form errors.

Methods: Four different neural networks SegNet, UNet, ResUNet and KiUNet were customized and trained for vertebra segmentation. To test and validate the workflow in clinical applications, an excised human lumbar vertebra was scanned via computed tomography (CT) and was reconstructed into 3D CAD models using the above four refined networks. A reverse engineering solution was proposed using a high-precision measuring robotic arm to obtain the original geometry of the excised vertebra as the gold standard.

Several 3D volumetric evaluation metrics and a finite element analysis (FEA) method were designed to show the model accuracy and model form errors for the geometries.

Results: The automatic segmentation neural networks achieved the best Dice score of 94.2% in the VerSe and CSI validation datasets. The accuracy of reconstructed models was quantified with the best 3D Dice index of 92.80%, 3D IoU of 86.56%, Hausdorff distance of 1.60 mm, and the heatmaps and histograms were used for error visualization. The FEA results of average von-Mises stress showed the impact of different geometries of the reconstructed vertebra on biomechanical results and reflected partial surface accuracy of the reconstructed vertebra under biomechanical loads with the closest difference of 1.924×10^4 Pa compared to the gold standard model.

Conclusion: In this work, a workflow of automatic subject-specific vertebra reconstruction methods was proposed while the errors in geometry and the corresponding stress distribution were quantified. Such errors should be considered when leveraging subject-specific modeling towards the development and improvement of treatments.

Abrégé

Contexte : Les modèles de colonne vertébrale actuels pour les modèles de banc analogiques, les plates-formes de navigation chirurgicale et de formation chirurgicale sont traditionnellement basés sur des modèles 3D produits à partir d'une base de données de polygones anatomiques du corps humain ou reconstruits à partir de données de numérisation spécifiques au sujet et étiquetées manuellement. Une méthode de reconstruction rapide et précise pour les modèles de colonne vertébrale spécifiques à un sujet est importante, car ces plates-formes sont souvent utilisées pour développer ou améliorer des traitements. Dans l'intervalle, les métriques d'évaluation de modèles 3D conventionnelles ne peuvent refléter que la précision globale du modèle statique et manquent de spécificité pour évaluer la précision du modèle sous différents angles.

Objectif : Proposer un flux de travail de méthode de reconstruction automatique des vertèbres spécifiques au sujet et quantifier la précision du modèle reconstruit et les erreurs de forme du modèle.

Méthodes : Quatre réseaux de neurones différents SegNet, UNet, ResUNet et KiUNet ont été personnalisés et entraînés pour la segmentation des vertèbres. Pour tester et valider le flux de travail dans l'application clinique, une vertèbre lombaire humaine excisée a été

scannée par tomodensitométrie (TDM) et a été reconstruite en modèles CAO 3D à l'aide des quatre réseaux raffinés ci-dessus. Une solution d'ingénierie inverse a été proposée en utilisant un bras robotique de mesure de haute précision pour obtenir la géométrie d'origine comme étalon-or. Plusieurs mesures d'évaluation volumétrique 3D et une méthode d'analyse par éléments finis (FEA) ont été conçues pour montrer la précision du modèle et les erreurs de forme du modèle pour les géométries.

Résultats : Les réseaux de neurones à segmentation automatique ont obtenu le meilleur score Dice de 94.2% dans les ensembles de données de validation VerSe et CSI. La précision des modèles reconstruits a été quantifiée avec le meilleur indice de dés 3D de 92.80%, IoU 3D de 86.56%, distance de Hausdorff de 1.60 mm, et les cartes thermiques et histogrammes ont été utilisés pour la visualisation des erreurs. Les résultats FEA de la contrainte moyenne de von-Mises ont montré l'impact de différentes géométries de la vertèbre reconstruite sur les résultats biomécaniques et reflétaient la précision partielle de la surface de la vertèbre reconstruite sous des charges biomécaniques avec la différence la plus proche de 1.924 $\times 10^4$ Pa par rapport au modèle étalon-or.

Conclusion : Dans cette étude, un flux de travail de méthodes de reconstruction automatique de vertèbres spécifiques à un sujet a été proposé tandis que les erreurs de géométrie et la distribution des contraintes correspondantes étaient quantifiées. De telles erreurs doivent être prises en compte lors de l'utilisation de la modélisation spécifique au sujet pour le développement et l'amélioration des traitements.

Acknowledgements

My master's program at McGill University is a special experience in my life, for I faced many unexpected conditions from the beginning of the pandemic of Covid-19 including the online courses and the quarantine, and also an unforgettable and warm experience, for I met many of you, my supervisors, my friends, my classmates and received so many kindness and helps. I wish to express my sincere appreciation to all of you, who make the cold city Montréal warm in my heart in the days to come.

I would like to first thank my research supervisor and mentor, Professor Mark Driscoll for providing continuous assistance throughout my research. Starting from the first year's online meetings to the second year's in-person meetings, my research topics become more and more clear and on the right track with the help of his professional guidance. Without his support and encouragement, I cannot overcome myself and get to where I am today and finish the research. I still remembered his suggestions on my future career which enlightened me and expanded my view out of the academic world. He and my co-supervisor Dr. Ahmed Aoude who supported my research with his expertise in the medical part are my most solid backings. I additionally like to thank my previous supervisor Jorge Angeles, from whom I learned a lot his rigorous attitude towards research.

I am also most grateful to everyone at Musculoskeletal Biomechanics Research Lab. Especially, I would like to thank Ibrahim El-Bojairami and Brittany Stott for introducing me to the world of finite element analysis with their professional knowledge and step by step teaching of the spine modeling in Ansys; Trevor Cotter for sharing his perfect McGill PPT templates which I used in my MECH 609 presentation; Swajan Paul and Siril Dukkipati for helping me have the CT scan of the excised human dried-out lumbar spine at the Research Institute of the McGill University Health Centre with Yongbiao Li and Antonio Aliaga's reliable and professional skills to generate a high-quality image data; also, my hardworking and never complaining partner - the 2018 MSI gaming laptop with a strong heart of GTX 1080 GPU that trained my deep learning models for many days and nights without a rest.

Finally, I would like to thank all those who helped support me outside of the research, my family and their video calls from thousands of miles away; my friends and those happy times when we fed little squirrels peanuts in winter; and many many others. Hope we can meet again someday in future!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As an important part of the musculoskeletal system, the spine plays a major role in mobility and supporting the human upper body. The 33 vertebrae that can be divided into five categories: cervical, thoracic, lumbar, sacrum and coccyx, have each corresponding functions. Due to the complex structure and biomechanics of spine, many spinal pathologies are under-diagnosed [2–4]. Therefore, the diagnosis, therapeutic method, and biomechanical analysis of the spine are challenging.

In modern spine surgery and surgical related applications, the subject-specific spine model is critical in various aspects to support diagnosis, preoperative planning, intraoperative navigation, surgical training platforms and biomechanical analysis. However, current spine models for analog bench models, surgical navigation and surgical training platforms generally come from anatomical human body dictionary database such as the BodyParts3D/Anatomography [5] or are reconstructed from time-consuming manual segmentation performed by professional radiologists on subject-specific scanning data [6]. Therefore, a quick and accurate reconstruction method for subject-specific spine models is

important as often such platforms are leveraged to develop or improve treatments.

Moreover, when quantifying the reconstructed model accuracy, current basic 3D model evaluation metrics in medical applications lack the specificity to evaluate the model accuracy from different perspectives. And these conventional metrics only compute the overall static model accuracy such as the mean deviation [7–10], the mean absolute distance [11–14] and the root mean square distance [15, 16], ignoring the critical partial surface accuracy - because even minor errors in the area between the intervertebral disc (IVD) and the vertebra will be magnified under spinal loads and hence greatly affect the results in biomechanics. Therefore, the model form assessment is required to be quantified from different perspectives both in geometry and under biomechanical loads. And such errors should be considered when leveraging subject-specific modeling towards the development and improvement of treatments.

To solve the above two main tasks, this research has the following objectives:

1. Design a workflow of automatic subject-specific vertebra reconstruction method.
    - Develop automatic vertebra segmentation algorithms using deep learning models.
2. Quantify the reconstructed model accuracy and model form errors.
    - Use reverse engineering to reconstruct a gold standard model for evaluation of the automatic reconstructed models in clinical applications.
    - Design different 3D model evaluation metrics and heatmaps/histogram for error visualization.

- Propose a finite element analysis (FEA) method to show stress distribution and to reflect the partial surface accuracy.

Chapter 2 is the literature review introducing the background of current subjuect-specific spine model applications, spine reconstruction methods and 3D model evaluation metrics; Chapter 3 includes the workflow of automatic vertebra reconstruction methods using four neural networks and tests the clinical performance on a human excised dried-out vertebra; Chapter 4 evaluated the overall static accuracy of reconstructed models using different 3D evaluation metrics; Chapter 5 proposed an FEA evaluation method to show stress distribution and to evaluate the partial surface accuracy; Chapter 6 includes the discussion and conclusion sections.

Chapter 2

Literature Review

2.1 Applications of Subject-specific Spine Models

The spine, also known as the vertebral column, plays a major role in mobility and supporting the human upper body. The 33 vertebrae that can be divided into five categories: cervical (C1 to C7), thoracic (T1 to T12), lumbar (L1 to L5), sacrum (S1 to S5) and coccyx (4 fused tailbones), have their corresponding functions. Different types of diseases such as spina bifida, spondylolisthesis, spondylolysis, spinal disc herniation, etc. and their therapeutic methods were being studied with the development of surgical procedures.

In modern spine surgery and surgical related applications, the subject-specific spine model is necessary for various aspects to support diagnosis, preoperative planning, intraoperative navigation, surgical training platforms and biomechanical analysis.

Due to the complex structure of spine and its multiple functions, spinal pathologies are often under-diagnosed [2–4]. In this situation, a subject-specific reconstructed three-dimensional (3D) model can provide a direct view of spatial morphology, facilitating the

recognition of spinal deformities such as kyphosis and scoliosis, the severity of vertebral fractures, etc. The reconstructed model helps an early diagnosis of these pathologies and effective treatments in time.

In recent years, computer assisted surgery (CAS) has greatly improved the operation accuracy in spine surgeries such as the pedicle screw placement [17] compared with the traditional free-hand surgeries. Based on the image-guided navigation system, the augmented reality (AR) technology was introduced in the CAS to further increase the efficiency and accuracy by merging the virtual information such as the subject-specific 3D spine hologram and virtual surgical path in the real environment. With the help of head-mounted display (HMD), the surgeons can see through the glasses the virtual information superimposed on the patient's surgical region so that they do not need to look away from the patient into another screen to check the preplanned surgical path and current surgical instrument's position. In 2013, Abe et al. first applied the AR navigation for vertebroplasty on the thoracolumbar spine [18]. Later, more applications in the pedicle screw fixation were proposed depending on the high performance of HMD, such as the *HoloLens*[1] [19–22]. In 2019, Wei et al. applied kyphoplasty for thoracolumbar spine using AR navigation [23]. In 2019, the first AR-based rod bending for lumbosacral spine was proposed by Wanivenhaus et al. [24] and was then improved by von Atzigen et al. [25]. In these real-time AR navigation system, a quickly-updated and accurate subject-specific 3D model will undisputedly and directly affect the final surgical precision and results.

Another practical use of subject-specific reconstructed model is in the virtual reality (VR) training platforms. The medical students, trainees and surgeons can use the

[1] An ergonomic, untethered self-contained holographic device developed by *Microsoft Corporation*.

high-fidelity VR simulator to improve the surgical techniques and validate the preoperative surgical plan in an immersive surgery environment with real-time interactions including visual, audible and haptic feedbacks. Gasco et al. proved the potential advantage of using the VR simulator for lumbar pedicle placement instructions by comparing two groups of medical students, in which the group using VR simulator outperformed the group using conventional visual instructions in all areas [26]. Shi et al. also assessed the validity of VR training platform on pedicle screw placement [27]. Gottschalk et al. tested the effect of VR simulator in the placement of cervical lateral mass screws using a blinded randomized control trial [28]. Halic et al. developed a VR and AR simulator for artificial cervical disc replacement, which was validated by five physicians [29]. In all these applications for VR training platforms, an accurate subject-specific reconstructed spine model is indispensable for creating the holograms in the simulator.

The subject-specific reconstructed 3D spine model can also serve as analog bench models for finite element analysis (FEA). As the biomechanical changes can reflect on spinal disease in either short term or in long term, such as the osteoporosis [30], the subject-specific FEA can not only analyse the biomechanics of spine, but also guide diagnostics and treatment [31]. Since the first 3D spine model was developed for pilot ejection studies in 1957 [32], more applications of spine FEA has appeared using numerical methods such as the damper and the spring-mass system [33]. The medical applications of FEA have focused on scoliosis [34–36], fractures [37–39], degenerative disc disease [40, 41] and osteoporosis [42–44], etc. Due to the diversity of interpersonal spine stiffness, subject-specific models can also be used for population-based analysis to better study the spine behavior among different ages [45]. With the development of 3D printing techniques, the subject-specific model can even be replicated

as 1:1 models for real biomechanical tests.

In clinical practice, most of the subject-specific spine models are reconstructed from image data of multiple medical imaging modalities, in which computed tomography (CT) scanning is the most preferred modality to study the vertebrae for its high contrast in bone to soft tissue. Due to the high spatial accuracy of CT image data, the subject-specific vertebrae include the 3D morphology, which can be observed in any slices from the axial, coronal or sagittal planes. Although the CT image data itself is enough to satisfy the requirements of some traditional image-guided spine surgical navigation methods, such as in the vertebral fusion surgery [46] and load analysis [47], most of the modern surgery and surgical applications still require a subject-specific 3D spine model reconstructed from the CT images.

However, current spine models for AR surgical navigation, VR surgical training platforms, and finite element analysis mainly come from two sides: (1) 3D models produced from anatomical human body dictionary database such as the BodyParts3D/Anatomography [5], where the spine polygon mesh files can be customized as per the research [48, 49]; (2) 3D models reconstructed from subject-specific imaging data of CT and magnetic resonance imaging (MRI) [6]. This requires manual segmentation performed by professional radiologists, which is accurate but subjective and very time-consuming for annotations. This research proposed an automatic workflow for subject-specific vertebra reconstruction with both high speed and accuracy.

2.2 Spine Reconstruction Methods

To reconstruct the spine model from subject-specific CT image data, the spine needs to be first segmented from bones and soft tissues at voxel level - the segmented image data, termed as mask or label, can delineate the exact boundaries and the interior area of the vertebra. After the segmentation, the Marching Cubes algorithm proposed by Lorensen et al. [50] and many of its variants such as the Flying Edges algorithm [51] can be used to render the surface as polygonal mesh on the basis of the segmented mask. The vertebral segmentation is a crucial step, not only because it indicates the spinal morphology and pathology, but also because the accuracy of the reconstructed model mainly depends on the segmentation quality.

2.2.1 Conventional Segmentation Algorithms

Due to the complex shape of the vertebra, the similar structure between adjacent vertebrae, and the spatial position of vertebrae, soft tissues and ribs, the vertebra segmentation is challenging. Traditionally, the vertebral segmentation was performed manually by experienced radiologists, which meets the requirement in accuracy but is subjective and time-consuming. Several semi-automatic segmentation algorithms were then widely used in clinical practice and the results were refined by professionals afterwards. One approach in early works is based on the image intensity: Kang et al. used region growth and adaptive thresholding for skeletal structures segmentation [52]. Lim et al. used the Willmore flow in level sets for spine segmentation [53] in 2014 and Hammernik et al. proposed a variational intensity framework [54] in 2015. Other algorithms include

model-based approaches such as the shape and pose statistic model by Rasoulian et al. [55], 3D superquadric model by Štern et al. [56], high-order Markov random fields by Kadoury et al. [57] and landmark-based shape representations by Ibragimov [58], etc. These approaches provided practical solutions for vertebral segmentation. However, the results still required refinement as the segmentation accuracy cannot satisfy the clinical demands. And meanwhile, most of the conventional segmentation algorithms took a long time for computation on every CT slices.

2.2.2 Automatic Segmentation Algorithms

More recently, with the advent of artificial intelligence, the supervised learning had an increase in the prevalence of end-to-end semantic segmentation tasks, including medical image segmentation. The accuracy of automatic spine segmentation has been further improved. In 2015, Suzan et al. employed multilayer perceptron (MLP) to locate the vertebral body and deformable shape modelling for segmentation [59]; Chu et al. [60] proposed a random forest classification method at voxel level for vertebra segmentation.

Due to the outstanding performance of convolution neural networks (CNN) in image processing tasks, different structures using convolution filters were proposed for spine segmentation. In 2017, Sekuboyina et al. used an MLP to locate the lumbar spine and then applied the UNet structure [61] for multi-class spine segmentation [62]. In 2019, Lessmann et al. designed an iterative fully CNN to segment the vertebra one after another using a sliding window [63]. More recently, Payer et al. adopted a coarse-to-fine segmentation method combining spine localization, labelling and segmentation using

Spatialconfiguration-Net and UNet [64]. The CNN-based UNet and its variants are now widely used in automatic spine segmentation for its high segmentation accuracy and quick inference speed. The models reconstructed from these segmentation results can also have better quality compared to conventional segmentaion results.

2.3 Spine Model Evaluation Metrics

Current 3D spine model evaluation metrics in medical applications usually include: the maximum deviation, the minimum deviation [65], the mean deviation [7–10], the mean absolute distance [11–14] and the root mean square distance [15, 16], etc. However, there are some limitations of the conventional metrics: (1) These metrics lack the specificity to evaluate the model accuracy from different perspectives and cannot reflect the association with the segmentation results; (2) These metrics only evaluate the overall static surface accuracy.

To overcome the first limitation, this research took the advantage of several evaluation metrics for 2D medical image segmentation tasks, such as the Dice coefficient [66], the Intersection over Union (IOU) [67] and the Hausdorff distance (HD), etc. These metrics have different properties, either sensitive to the boundaries or to the interior area of the segmented mask. To make full use of these properties, the 2D image evaluation metrics were extended to 3D volumetric evaluation metrics in this research. Based on the Hausdorff distance, the heatmaps and histograms were also used for error visualization. More details will be presented in Section 4.1.

To solve the second limitation, this research introduced an FEA method for model form

assessment to explore the impact of different geometries of the reconstructed vertebra on biomechanical results using average von-Mises stresses so as to evaluate the critical partial surface accuracy. The FEA evaluation methods can compute the deformation and stress distribution of the model under various biomechanical scenarios, where the regional surface accuracy is more critical than the overall model accuracy. Because even minor errors in the area between the intervertebral disc and the vertebra will be magnified under spinal loads and hence greatly affect the biomechanical results. The FEA provides a different approach for 3D model evaluation, not only analysing the biomechanical results but also reflecting the critical partial surface accuracy of the reconstructed vertebra.

Chapter 3

Automatic Vertebra Reconstruction

In this chapter, four different neural networks were first customized for automatic vertebra segmentation. Then, two public spine datasets were used for training and validating the segmentation accuracy. Several practical strategies were applied for facilitating the training procedures. The experiment results showed the training details and the segmentation results. Finally, a human excised dried-out vertebra was introduced to test the automatic segmentation performance in clinical applications and was then reconstructed into surface mesh model using the segmented masks from the above four refined networks. The originial geometry of the dried-out vertebra which served as the gold standard was obtained using reverse engineering.

3.1 Automatic Vertebra Segmentation Structures

In this section, four deep learning models based on the encoder-decoder architecture: SegNet, UNet, ResUNet, and KiUNet were modified and customized to automatically

segment the vertebra from the CT image data. All the four neural networks can be trained to extract 3D volume information from the input tensor using 3D convolution filters and output the corresponding segmented vertebra masks - the annotations of the vertebrae.

3.1.1 SegNet

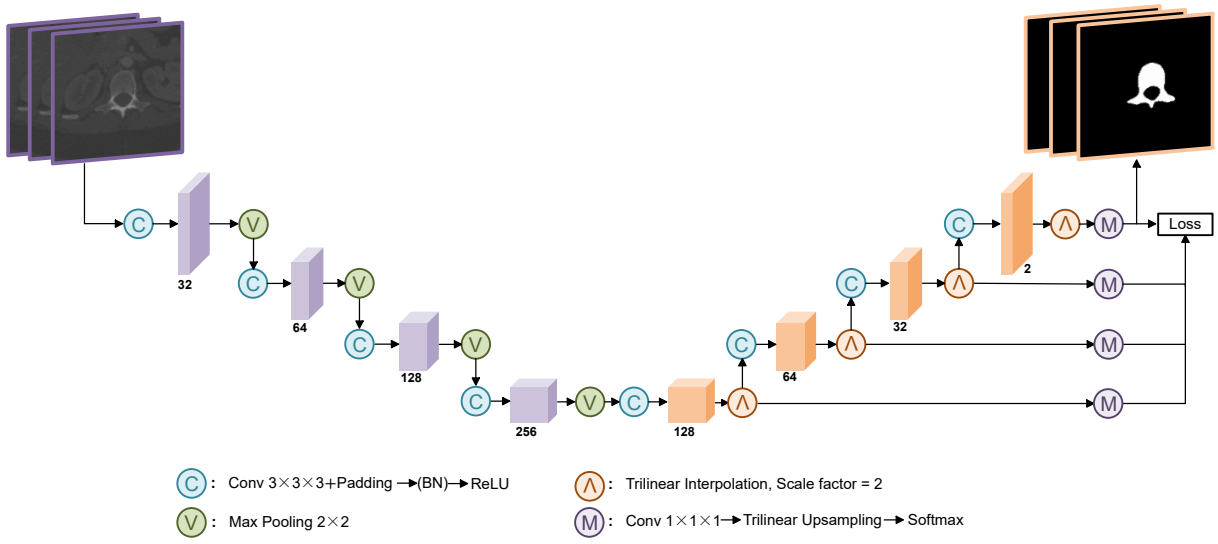


Figure 3.1: Schematic diagram of a custom 3D SegNet neural network structure.

In 2015, Vijay et al. proposed a fully convolutional neural network using an encoder-decoder architecture for semantic segmentation, the SegNet [68], which achieved better segmentation results than other architectures with less inference time and less memory usage on both the road scenes and indoor scenes datasets. The SegNet includes an encoder network, topologically same as the convolutional layers in the VGG network [69]; a decoder network, to restore the size of low-resolution encoder feature maps back to the original input image size; and a final classification layer using softmax classifier [70], a non-linear activation function for pixel-wise segmentation. In the encoder, the RGB image tensor is first input into a convolutional layer to learn hidden features, producing a set of

feature maps. Following the convolutional layer, the output feature maps are applied the pixel-wise rectified linear activation function (ReLU) [71]: $max(0, x)$, for non-linearity. Then, a max pooling layer [72] with a 2×2 window and a stride of 2 is used for downsampling, reducing the dimension of feature maps to half by selecting the maximum value of pixels in every 2×2 windows. The max pooling layer also helps provide translation invariance against small spatial shifts of the features. The downsampled feature maps are then input into the next convolutional layer and the above steps are repeated to learn more compact hidden hierarchical features. In the decoder, the max pooling indices in the encoder are reused for upsampling and the sparse upsampled feature maps are convolved with convolution filters to generate dense feature maps. The final softmax layer can be trained to produce an N-channel image of probabilities for N segmentation classes, with the maximum probability at each pixel corresponding to the segmentation results of that class.

Based on the original SegNet structure, this research proposed a custom 3D SegNet structure for vertebra segmentation, as is shown in the schematic diagram Fig. 3.1. Different from the initial SegNet, the custom 3D SegNet made the following modifications and improvements for spine segmentation tasks:

(1) The architecture was modified for 3D convolution using 3D convolutional filters instead of 2D convolution. Unlike single colorful images, medical image data using tomographic imaging techniques is comprised of stacked images usually larger than 100 continuous slices. Hence, the dimension of the input image tensor would be $Height \times Weight \times Depth$. Due to this property of medical images, the original 2D convolution was extended to 3D convolution which has a larger volume receptive field to

better use the 3D volume information from the input tensor. In Fig. 3.1, the number below each tensor denotes the number of channels, which equals the number of convolutional filters. The $3 \times 3 \times 3$ convolutional filters were used in the 3D convolution with a stride of 1 and padding of 1 to keep the output tensor size the same before convolutions.

(2) A deep supervision mechanism using auxiliary loss [73] at the end of each stage in the decoder was introduced to facilitate the final loss backpropagating to early layers and to improve the segmentation accuracy. In deep neural networks, the gradients vanishing problem [74, 75] always occurs during backpropagation, hampering the weights being updated. This problem is even more severe in the 3D convolutional neural network which uses the voxel-wise classification for segmentation. The backpropagation in deep supervision branches can effectively alleviate the vanishing gradients and help accelerate the convergence, by deriving the gradients directly from these branches.

Specifically, three additional lower-level feature maps were first convolved with $1 \times 1 \times 1$ convolutional filters, reducing the tensor channels to the number of labels which is 2 in our case (foreground and background). Secondly, they were all upscaled to the size of input image tensor size using trilinear interpolation with scale factors of 2, 4 and 8 respectively. Then, the three low-level updated feature maps applied a voxel-wise softmax activation function for classification. Finally, all three outputs as well as the output from the end of the decoder were compared with the ground truth mask respectively using the loss function in Section 3.3.1 to calculate the total segmentation error. The final auxiliary loss was comprised of the following two parts:

$$Loss_{total} = Loss_{main\ path} + \omega \cdot (loss_0 + loss_1 + loss_2) \tag{3.1}$$

where $loss_{0,1,2}$ is from three low-level feature maps compared with the ground truth mask, ω is a balancing weight, which will be decayed over training to prevent the deep supervision mechanism from affecting the segmentation results negatively in the late stage of training. This is because the final segmentation result only comes from the output at the end of decoder rather than from the branches.

Due to the hierarchical structure in the decoder, the branches extended from lower-level layers have different size of receptive fields that helps the model learn from multiscale context information to improve the segmentation accuracy.

(3) Other differences include: The batch normalization layer [76] after the convolutional layer is optional to use for faster and more stable convergence. The trilinear interpolation was used in the decoder after each convolutional layer with a scale factor of 2 to double the size of the output tensor.

The output tensor at the end of the decoder has 2 channels and was then mapped to the softmax layer for classification. The pixel with the larger value in the two channels was classified to the foreground - the vertebra, and the pixel with the smaller value was set to the black background.

3.1.2 UNet

Based on the fully convolutional networks for semantic segmentation [77], Ronneberger et al. proposed a popular neural network structure termed UNet [61] for biomedical image segmentation. It achieved the best Intersection over Union (IOU) score of 77.5% in the cell segmentation task challenge in 2015. As its name suggests, the UNet also includes a

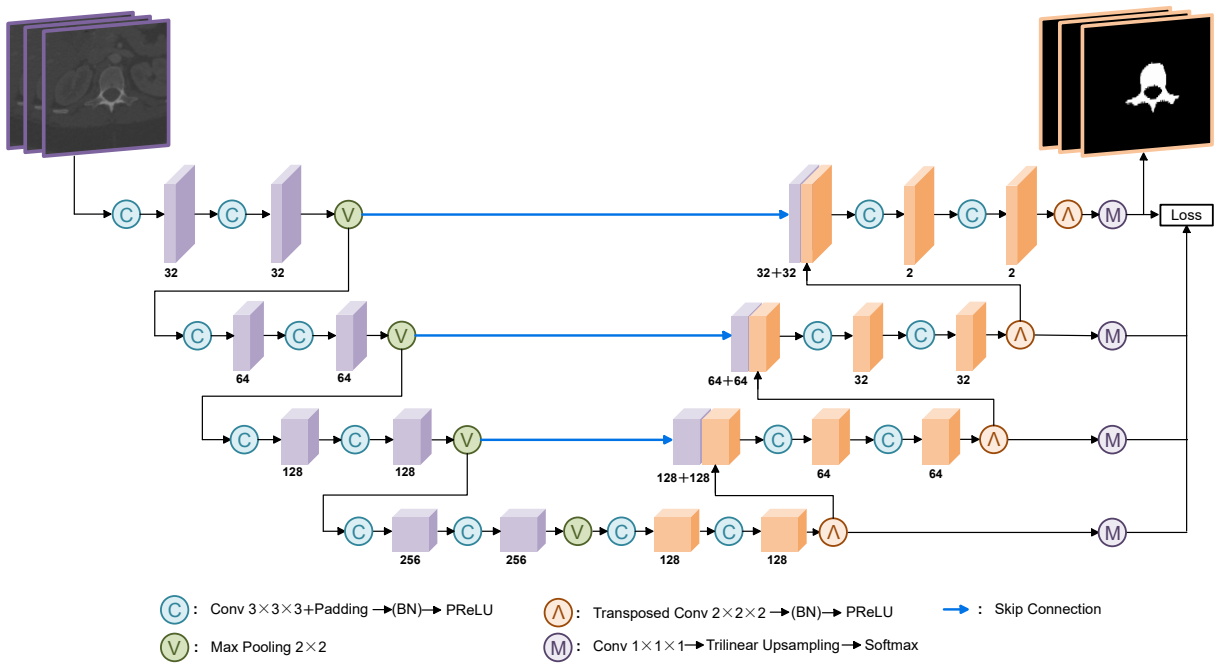


Figure 3.2: Schematic diagram of a custom 3D UNet neural network structure.

contracting path (encoder) and a symmetric expanding path (decoder), yielding a U-shaped architecture. However, unlike the computationally efficient SegNet, the UNet applied more successive convolutional layers in the hierarchical structures. Another big difference is that the U-Net adopted skip connections between the encoder and the decoder, facilitating the transmission of local features from the encoder to decoder. Specifically, the feature maps in the encoder will be concatenated to the corresponding upsampling tensors in the decoder. The concatenated feature maps with a large number of feature channels can thus provide the lost information on the encoder side because of downsampling to higher layers. The skip connections also help mitigate the gradients vanishing problem by backpropagating through these shortcuts.

Based on the original UNet, Çiçek et al. proposed the 3D UNet [78] for dense volumetric segmentation by replacing the 2D operations with 3D counterparts. This research made a

further step and proposed a custom 3D UNet for vertebra segmentation, as is shown in the schematic diagram Fig. 3.2. Compared with the 2D UNet, the custom 3D UNet had the following modifications and improvements:

(1) The architecture was extended to 3D operations and more successive 3D convolutional layers than in the SegNet were used to extract hidden features. In the decoder, 3D transposed convolutional layers were applied for upsampling with $2 \times 2 \times 2$ convolutional filters and strides of 2, upscaling the feature maps by a factor of 2. The advantage of using transposed convolution is that the transposed convolutional filters can be trained to learn how to upsample the segmentation map and double the resolution, rather than simple trilinear interpolation in pixels.

(2) The deep supervision mechanism was applied in the same way as in the custom 3D SegNet to help backpropagate and to improve the segmentation accuracy.

(3) The regular activation function, ReLU was replaced by the Parametric Rectified Linear Unit (PReLU) [79] for adaptive nonlinearities after the convolutional layers. As is shown in Fig. 3.3, the coefficient α in the negative part of PReLU can be adaptively learned with little extra computational cost. According to the findings from He et al. [79], the initial layers try to detect features such as edges and textures, suggesting the model tends to become linear with α greater than 0; on the contrary, deeper layers are more "non-linear" and discriminative, requiring a smaller α. When the learnable coefficient α is small enough, the PReLU also has the property of Leaky ReLU [80] to avoid zero gradients.

The output tensor at the end of the decoder has 2 channels and was then mapped to the softmax layer for classification. The pixel with the larger value in the two channels was classified to the foreground - the vertebra, and the pixel with the smaller value was set to

the black background.

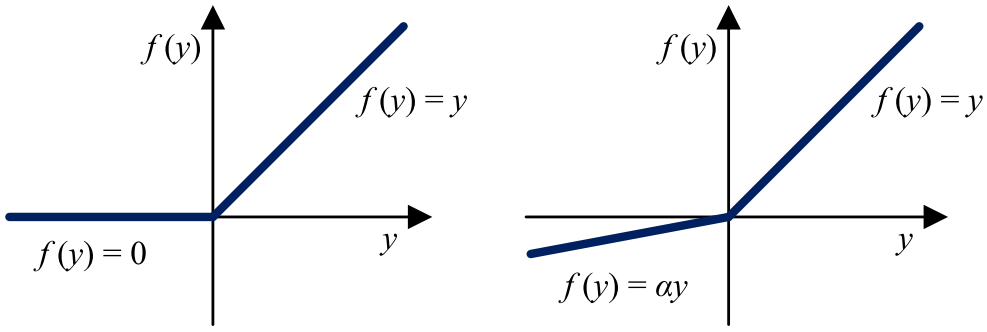


Figure 3.3: Activation functions of ReLU (left) and PReLU (right) with adaptive coefficient in the negative part.

3.1.3 ResUNet

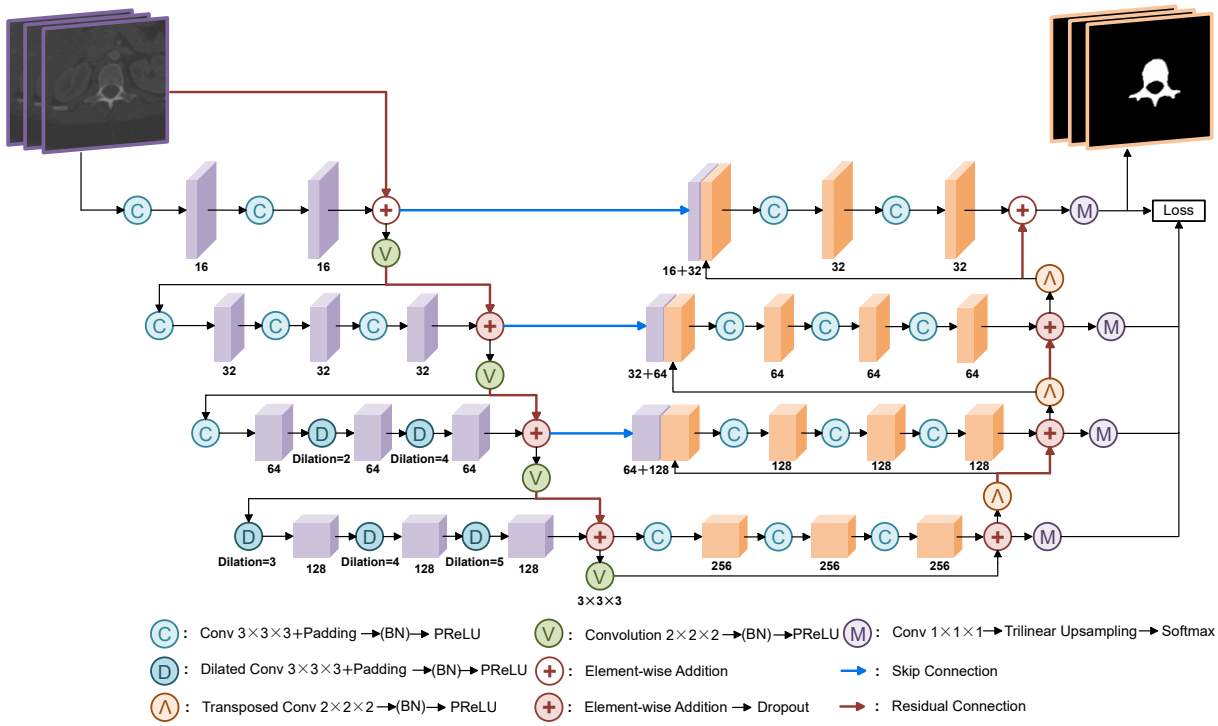


Figure 3.4: Schematic diagram of a custom 3DResUNet neural network structure.

The Deep Residual UNet (ResUNet) was first proposed by Zhang et al. [81] to extract roads from aerial images. The ResUNet takes advantage of both the UNet and ResNet [82].

The ResNet, with its residual blocks, was proposed by He et al. [82] to ease training and address the degradation problem. Intuitively, a deeper model can learn from high-level features and achieve better results than "shallow" models; however, during training, He et al. found that when the deep model was about to converge, the accuracy tended to saturate and then degraded soon - because the information from shallow layers becomes more difficult to propagate and therefore gets lost. The residual connection was then designed to address the degradation problem by creating a shortcut to directly send the information from shallow layers to deep layers. Specifically in the ResUNet, after the feature maps are fed into the downsampling or upsampling layer, the output tensor will be input into next deeper hierarchical path in one branch; and in another branch, it will be added directly to the output feature maps in this path element-wisely, as they have the same dimension.

Based on the original ResUNet, Zhang et al. proposed a dial/hybrid cascade 3D ResUNet [83] for liver and tumor segmentation. In light of this structure, a custom 3D ResUNet architecture based on the 2D ResUNet was used for vertebra segmentation, as is shown in Fig. 3.4. The red lines in the schematic diagram are the residual connections in the encoder and decoder. Other improvements and modifications compared with the 2D ResUNet include:

(1) The architecture was extended to 3D operations and more successive 3D convolutional layers than in the UNet were used in both the encoder and decoder to extract compact representations.

(2) The 3D dilated convolutions were used in deep layers in the encoder to enlarge the receptive field (or the filter size) without increasing parameters. Fig. [1] shows a 3×3 convolution kernel convolving over a 7×7 feature map with a dilation factor of 2. In

convolutional neural networks, the receptive field plays a vital role as it determines the capability of convolution filters to perceive the spatial connectivity in the input feature maps. In the encoder, the more downsampling layers are used, the more dilated information will get lost if the normal convolution filter is still used over the compact feature maps. Therefore, in the last two stages of the encoder in the custom 3D ResUNet, the normal convolution operations were replaced by dilated convolutions to counteract this side effect. However, Wang et al. [84] pointed out that continuous dilated convolution with the same dilation factor would cause checkerboard effects - the receptive field only covers an area with checkerboard patterns and the sparse feature map will then lose part of local information permanently. So they proposed a hybrid dilated convolution, which is used in this custom 3D ResUNet: the dilation factors in the last two stages were 2, 4 and 3, 4, 5 respectively.

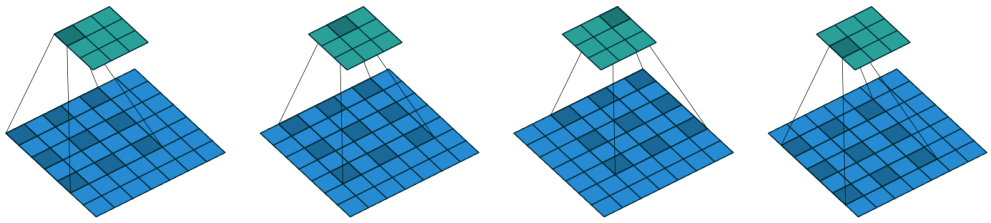


Figure 3.5: The dilated convolution with a 3×3 convolutional filter and a dilation factor of 2 [1].

(3) The dropout was applied after the summation of residual connections except for the first and last one, to prevent overfitting problems. Details can be found in Section 3.3.5.

(4) The same deep supervision mechanism was used to help backpropagate and to improve the segmentation accuracy.

The output tensor at the end of the decoder has 2 channels and was then mapped to the softmax layer for classification. The pixel with the larger value in the two channels was

classified to the foreground - the vertebra, and the pixel with the smaller value was set to the black background.

3.1.4 KiUNet

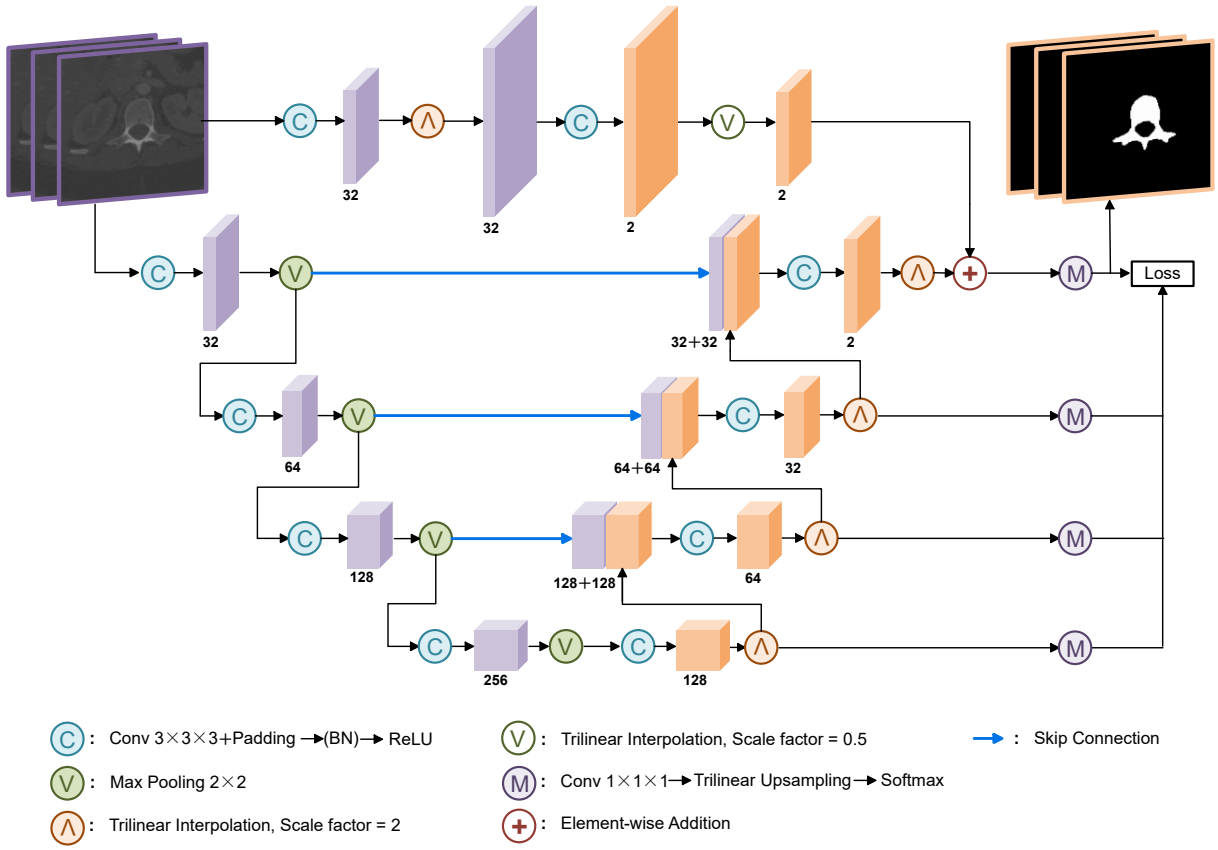


Figure 3.6: Schematic diagram of a custom light version of 3D KiUNet neural network structure.

The UNet architecture and its variants have performed quite well in segmentation tasks based on the backbone of the encoder-decoder structure. In deep layers of the encoder, the model can learn high-level features from the compact representations with an increase in the receptive field size. However, on the other side, the model will inevitably miss low-level features such as the small structures and boundaries of the vertebra. To solve this drawback,

Valanarasu et al. proposed the KiUNet [85], which combines two networks: the Kite-Net (KiNet) and the UNet. The KiNet, which is an overcomplete convolutional network, can map the input image to a higher dimension so that the model can learn to extract small structures and fine details. Moreover, to exploit the full capacity of the two networks, Valanarasu et al. designed a cross residual feature block (CRFB) to combine the features from the two networks at different stages, which helps add complementary features during training.

In light of the original complex KiUNet structure which requires high computational resources, a custom light version of 3D KiUNet was proposed in this research for vertebra segmentation with a simpler structure and high accuracy. The custom 3D light KiUNet includes two main networks, as is shown in Fig. 3.6:

(1) The above one is the KiNet, an overcomplete architecture to map the inputs to higher dimensions so that the model can focus more on low-level features, such as the fine details of small structures and boundaries of the vertebra. More specifically, the input image tensor was first fed into a $3 \times 3 \times 3$ convolutional layer with 32 convolutional filters, and was then upsampled by a scale factor of 2 using trilinear interpolation. The output feature maps were fed into another convolutional layer with only 2 convolutional filters, reducing the number of channels to the one of final output segmentation maps. A max pooling layer was followed to downsample the feature maps back to the resolution of the input image tensor.

(2) The bottom network is the UNet, an undercomplete architecture to map the inputs to lower dimensions so that the model can focus on high-level features, such as the large structures like the main body of the vertebra. The UNet was simplified with only one convolutional layer in each stage of the encoder and decoder to reduce memory usage.

The output tensors with 2 channels from KiNet and UNet were then applied element-

wise summation and mapped to the softmax layer for classification. The pixel with the larger value in the two channels was classified to the foreground - the vertebra, and the pixel with the smaller value was set to the black background.

3.2 Datasets

A dataset is a collection of data pieces with certain distributions, from which a deep neural network can be trained to learn the features and then output the predicted features. The dataset can be also used for validation during training or for tests after training to evaluate the generalization performance of the model on unseen data.

In supervised learning tasks such as the medical image segmentation, each image data is associated with a label, from which the deep learning model can extract an optimal feature representation from the input image. Therefore, the quality of the datasets, including (1) the amount and variety of the dataset to improve generalization ability (2) accurate labelled masks to ensure the segmentation accuracy, will greatly affect the training results of deep learning models.

Based on the above two principles, two public spine datasets were used in this research. The first is from *Large Scale Vertebrae Segmentation Challenge (VerSe)* held in conjunction with *2019 Medical Image Computing and Computer Assisted Intervention (MICCAI)* conference. The *VerSe* dataset contains 160 multi-detector CT (MDCT) scans of 141 patients with a mean age of $\sim 59(\pm 17)$ years and altogether 1725 vertebrae were annotated at voxel level by a human-machine hybrid algorithm [86]. The MDCT scans from the *VerSe* dataset include but are not limited to fractured vertebrae, metallic

implants, cemented vertebrae, transitional vertebrae and noisy scans. The second dataset is from the test set of 2014 *MICCAI* Workshop on *Computational Spine Imaging (CSI)* [87], which is publicly available on SpineWeb[1]. The *CSI* dataset includes 10 CT scans covering the entire thoracic and lumbar spine. Five cases were from healthy young adults (20-34 years, mean 27 years) and the other five had at least one vertebral compression fracture from an osteoporotic cohort (59-82 years, mean 73 years). The annotations of *CSI* were manually labelled and refined by a medical fellow and a research fellow.

The two datasets (160 *VerSe* + 10 *CSI*) as well as the corresponding labels were randomly split into training set (128 *VerSe* + 8 *CSI*) and validation set (32 *VerSe* + 2 *CSI*), where the validation set was not involved in training and was only used to compute the validation Dice score and validation loss to test the generalization ability.

As the two CT datasets were scanned by multiple CT scanners with different manufacturers (GE, Siemens, Toshiba, etc) and used diverse scan settings, the in-plane resolution, the slice thickness and the Hounsfield unit (HU) values were mostly different, which will disturb the training process to learn from the right distribution of features. Hence, the two datasets require preprocessing before training and validation. The image intensity was all thresholded between -2500 and 3500 to maintain the main features. Due to the different pixel spacing and slice thickness, the voxel spacing was resampled to $1\times1\times1$ *mm*. To maintain the real scale of the vertebra, the image size was restored back to its original size according to the following equation:

$$New\ Size = Original\ Size \times Original\ Spacing\ /\ New\ Spacing \tag{3.2}$$

[1]spineweb.digitalimaginggroup.ca

As the dimensions of the CT images were also different from each other, the image size of every CT slice was expanded to times of 32 by padding the input tensor using replication of the input boundary, creating a feasible size for the convolutional layers in downsampling and upsampling path.

3.3 Training Strategies

This section elaborates on the choice and design of several training strategies to achieve better training results, including the loss function to compute the error, the optimizer to minimize the error, weights initialization methods, early stopping, dropout strategy to prevent overfitting problems and data augmentation to expand the dataset.

3.3.1 Loss Function

The loss function plays a vital role in instigating the learning process of neural networks because deep learning algorithms use different optimization methods such as stochastic gradient descents in back propagation [88] to update the model weights and minimize the loss function. The choice of the loss function is therefore extremely important; the principle to design a loss function is to effectively represent the error between the ground truth and the predicted outputs.

In medical imaging, the pixels grouped together in an image are defined as different elements. The medical image segmentation, as the semantic image segmentation task, aims to classify these elements at pixel level. To compare the accuracy of the segmented label with the ground truth, the Dice index is widely used as an evaluation metric in medical

imaging since 1994 [89]. The Dice index was later adapted to Dice loss by Carole H Sudre et al. in 2017 [90] as a loss function to train a deep learning model for the first time. More details were presented in Section 4.1.1.

Based on the Dice index, Wong et al. [91] proposed an exponential logarithmic loss, combining the Dice score and cross entropy loss to focus on less accurately predicted cases. For simplicity, only the exponential logarithmic Dice loss was adapted in this research, as shown in Equation 3.3.

$$L_{Dice} = \mathbf{E}\left[(-\ln(Dice + \epsilon)^{\gamma})\right] \tag{3.3}$$

$$\text{with } Dice = \frac{2(\Sigma_i \delta_{\mathbf{xy}}(i)) + smooth}{\Sigma_i \mathbf{x} + \Sigma_i \mathbf{y} + smooth} \tag{3.4}$$

where i denotes all the coordinates of pixels in a single slice of CT image. $\mathbf{x}$ and $\mathbf{y}$ are the value of the predicted label and ground truth label at i, either 0 for the background or 1 for the foreground elements. $\mathbf{E}[\cdot]$ is the mean value of Dice score with respect to all slices of an CT image data. $\delta_{\mathbf{xy}}(i)$ is the Kronecker delta: when $\mathbf{x} = \mathbf{y}$ at i, $\delta_{\mathbf{xy}}(i) = 1$, otherwise, $\delta_{\mathbf{xy}}(i) = 0$. ϵ is 10^{-5} to prevent invalid logarithm. The *smooth* parameter is the pseudocount for Laplace smoothing to prevent overfitting, which is 1 in this research. The γ in Equation 3.3 can further control the nonlinearities of the losses. According to Wong et al., when $\gamma = 3$, the loss function has an inflection point around $Dice = 0.5$, achieving a good balance between both low and high prediction accuracy samples.

3.3.2 Optimization Techniques

To minimize the loss function during training, an efficient optimization method can not only find the global minimum expeditiously with little memory usage but can also update well in poor conditions: a situation when the function $f(\theta)$ will change rapidly with respect to small changes of inputs.

Gradient-based optimization is a quite useful iterative optimization solution as the derivative of the loss function in every step indicates the next direction to update the weights θ. The first ideal choice is using the second-order gradients, also known as the Newton's method, which takes aggressive and short steps in directions of the curvatures to reach the global minimum. However, the drawback is that it generally requires the computation or estimation of the Hessian matrix (and the inverse Hessian matrix) of $f(\theta)$, requiring exceeded memory capacity. For example, a deep learning model with one million parameters has to compute a Hessian matrix of size [1,000,000×1,000,000], using around 3725 GB memory [92], which is impractical in neural network applications.

The second choice is the first-order gradient-based stochastic gradient descent (SGD) methods [93]. It is widely used for its simplicity and effectiveness as the SGD always searches for the steepest descent to update the weights, as shown in Equation 3.5.

$$\theta_t = \theta_{t-1} - lr \cdot \nabla_{\theta_{t-1}} f(\theta_{t-1}) \tag{3.5}$$

where $\nabla_\theta f(\theta)$ is the gradient of loss function at θ and lr is the learning rate, a hyperparameter of the stepsize in every updates. However, when $f'(\theta) = 0$, in which the gradients cannot provide further information, the points are called critical points. Except

for the ideal situation where the global minimum point is successfully found, other situations may include: a maximum point; a saddle point, which is usually surrounded by flat regions, difficult for updating because of small gradients; a local minimum point, which is lower than neighbor points but higher than the global minimum.

To jump out of critical points, adding a momentum term [94] can help accelerate the SGD by dampening the oscillations. From a physical perspective, when a ball rolls down a hill, it accumulates momentum to come across the local minimum. As is shown in Equation 3.6, the $\mu \cdot v$ is the momentum term where the hyperparameter μ (typically $\mu = 0.9$) can be interpreted as the "coefficient of friction" to stop the ball at the bottom of the hill:

$$\begin{aligned} v_t &= \mu \cdot v_{t-1} + lr \cdot \nabla_{\theta_{t-1}} f(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} - v_t \end{aligned} \tag{3.6}$$

Another critical problem in training neural networks is to anneal the learning rate over time, especially in poor conditions. A large learning rate in the beginning helps accelerate the rate of convergence to save time while a small learning rate helps reach the best position. Duchi et al. proposed Adagrad [95] in 2011 which includes an adaptive learning rate: the higher the gradients in the last iteration, the more the learning rate will be reduced.

$$\begin{aligned} v_t &= v_{t-1} + \nabla^2_{\theta_{t-1}} f(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} - lr \cdot \frac{\nabla_{\theta_{t-1}} f(\theta_{t-1})}{\sqrt{v_t} + \epsilon} \end{aligned} \tag{3.7}$$

where smooth term $\epsilon = 10^{-8}$ avoids division by 0. However, as the Adagrad used the accumulations of gradients in previous iterations, it usually stops too early. In 2012, Tieleman et al. proposed RMSProp [96] that used a moving average of square gradients instead:

$$
\begin{aligned}
v_t &= \beta \cdot v_{t-1} + (1-\beta) \cdot \nabla^2_{\theta_{t-1}} f(\theta_{t-1}) \\
\theta_t &= \theta_{t-1} - lr \cdot \frac{\nabla_{\theta_{t-1}} f(\theta_{t-1})}{\sqrt{v_t} + \epsilon}
\end{aligned}
\tag{3.8}
$$

where $\beta = 0.9$ is the hyperparameter of decay rate to equalize the effect of aggressive decreasing learning rate in Adagrad.

Combining the advantages of momentum and RMSProp, Diederik et al. proposed the Adam Optimizer [97], a first-order gradient-based optimization of stochastic objective functions in 2015. This research used the Adam optimizer as described in Algorithm 1. It is worth noting that as m_0 and v_0 are initilized at 0, they are also biased towards 0 during initial time. Therefore, a bias correction mechanism was applied to $\hat{m}_t$ and $\hat{v}_t$ to counteract the biases.

Algorithm 1 Adam Optimizer. Default $\beta_1 = 0.9$, $\beta_2 = 0.9999$ and $\epsilon = 10^{-8}$. g_t^2 is the elementwise square. t in $\beta_{1,2}^t$ denotes the exponent.

Require: lr: Learning rate
Require: β_1, $\beta_2 \in [0, 1)$: Exponential decay rates for moment estimates
Require: $f(\theta)$: Loss function w.r.t. weights θ
Require: θ_0: Initial weights

$m_0 \leftarrow 0$ ▷ Initial first moment variable
$v_0 \leftarrow 0$ ▷ Initial second moment variable
$t \leftarrow 0$ ▷ Initial time step
while training continues **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$ ▷ Compute the gradients of loss function at t
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ ▷ Update biased first moment
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ ▷ Update biased second moment
    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ ▷ Update bias-corrected moment
    $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ ▷ Update bias-corrected moment
    $\theta_t \leftarrow \theta_{t-1} - lr \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ ▷ Update weights
return θ_t ▷ Return updated parameters

3.3.3 Weights Initialization

Before the neural network starts training, the weights of the model are usually given a certain distribution to better prevent gradients vanish problem during back propagation. Early CNNs were mostly initialized by normal distribution (Gaussian distribution) [98]. However, in more recent deep neural network structures such as VGG [69], the gradients will easily get vanished during back propagation [99] so the model has difficulties converging. This is because the variance of the activation results will decrease after each layer so that the gradients will also gradually vanish during back propagation in deep layers, making it difficult to update the weights.

In 2010, Glorot et al. proposed the Xavier initialization [100] with scaled uniform distribution to solve this problem. The Xavier initialization can be denoted as:

$$W_{ij} \sim U\left[-\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}\right] \tag{3.9}$$

where W_{ij} is the weight and N is the number of input parameters. All the biases are set as 0. The Xavier initialization was designed to successfully prevent the variance of the back-propagated gradients from decreasing; however, it works well only with linear activation functions, but not with rectified linear unit activation functions.

When proposing the Parametric Rectified Linear Unit (PReLU), He et al. also derived the Kaiming initialization [79] that particularly takes the rectifier nonlinearities into consideration. Intuitively, as the rectified linear unit clamps almost half the output to 0, the mean value will be doubled, and so as the standard deviation (STD). Therefore, to keep a zero-mean Gaussian distribution, the STD was changed from $\frac{1}{\sqrt{N}}$ to $\sqrt{\frac{2}{N}}$. A rigorous

proof can be found in the paper [79]. In experiments, the Kaiming initialization allows neural networks with 30 convolution layers to converge while the Xavier initialization can not. And their PReLU networks with Kaiming initialization achieved a 26% relative improvement over the GoogLeNet [101] and surpassed human-level performance on visual recognition challenge first time in history. In this research, the Kaiming initialization was used for all four deep learning models.

3.3.4 Early Stopping

In this chapter, all four neural networks were applied early stopping to avoid overfitting problems during training.

Early in 1995, Tom Dietterich has pointed out a central problem in supervised learning: as the machine learning model is trained and evaluated based on the training data, it can well predict the observations in the training data after training; however, the overfitting problem will then occur when a model is too well fit with the training data, incurring a large generalization error, that is, performing poorly on new, previously unseen data.

The regularization was then introduced to solve this contradiction. Girosi et al. [102] studied the application of the regularization term in neural networks in 1995, indicating that it actually imposed a smoothness constraint on the learning model, such as the L^2 regularization, also known as weight decay or Tikhonov regularization. The L^2 regularization is added to the loss function as a parameter norm penalty in the form of $\Omega(\theta) = \frac{1}{2}\|\omega\|_2^2$, where θ represents all of the parameters and ω represents the parameters affected by a norm penalty. Assume the optimal value of weights for the loss function $J(\omega)$ is $\omega^* = \arg\min_\omega J(\omega)$,

a quadratic loss function with L^2 regularization for a linear regression model can be denoted as the approximation $\hat{J}(\theta)$:

$$\hat{J}(\theta) = J(\omega^*) + \frac{1}{2}(\omega - \omega^*)H(\omega - \omega^*) \tag{3.10}$$

where H is the Hessian matrix of J. As shown in Fig. 3.7a, the gradient vanishes at ω^*, which is the minimum of the loss function. The dotted circles are the equal values of regularizer L^2 while the solid circles are the equal values of the unregularized loss function. The equilibrium point $\tilde{\omega}$ is the new optimal weights for the loss function with regularization term. As can be perceived that the weights $\tilde{\omega}$ are generally smaller than ω^*, the model will be able to cope with inputs of high variance, compared with larger weights that high variance inputs will greatly affect the results.

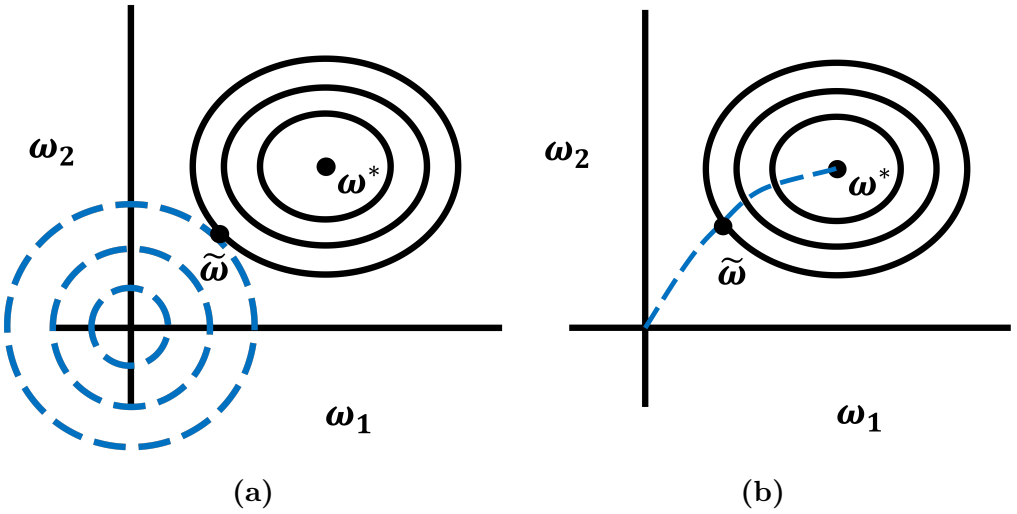


Figure 3.7: An illustration of equal values of quadratic loss function with regularization (a) L^2 regularization (b) Early stopping

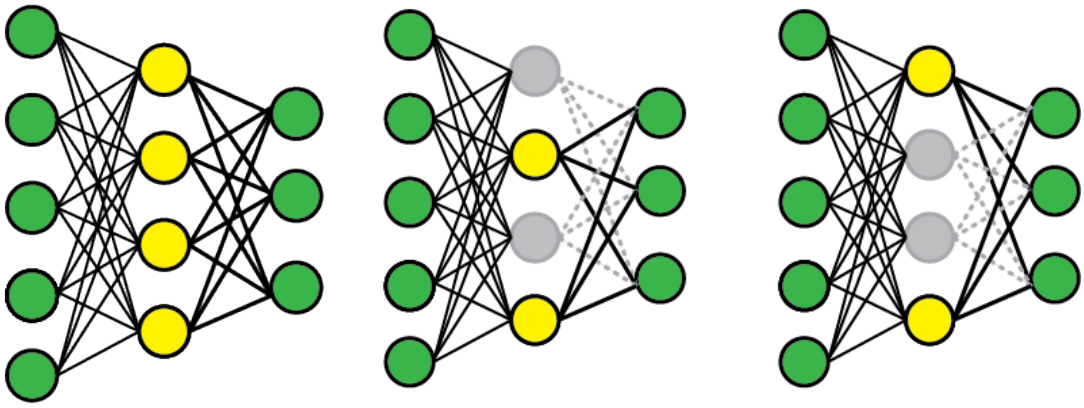
Early stopping, as its name suggests, is to stop training before finishing the targeted training epochs if the results did not improve. Early stopping is widely used for its simplicity

and effectiveness to reduce the cost of computation. Fig. 3.7b shows how early stopping functions as a regularizer in a quadratic loss function: during gradient descent, early stopping restricts the iterations of optimization algorithms to stop at an earlier point $\tilde{\omega}$, which has a similar effect of L2 regularization. But early stopping is unobtrusive and does not change any training procedures - it just simply stops the training. However, the number of epochs in early stopping remains a hyperparameter. In ideal situations, when the overfitting is observed in the validation loss curve when the loss is increasing instead of decreasing during training, the early stopping should then be applied. But the real validation loss curve has more fluctuations, which is not typical to estimate the overfitting occurs. Prechelt [103] in 2012 concluded a tradeoff between training time and generalization, presenting a way to solve this problem.

3.3.5 Dropout

In deep neural network structures such as the ResUNet in Section 3.1.3, the dropout, proposed by Srivastava et al. in 2012 [104, 105] was used to prevent overfitting by reducing co-adaptation of units in a neural network. In other words, part of the units is randomly omitted and their output will be clamped to 0 so that the rest part of units is able to independently learn features from the input with less dependence on other neurons.

Fig. 3.8 shows an example of how dropout functions in a 1-hidden-layer feedforward network with a 0.5 drop rate. In Fig. 3.8b, half of the units are disabled randomly and half of the units are selected to connect to the subsequent layer during training. When testing, all the units become active again and the learned weights are halved for compensation.



(a) Original neural net **(b)** After dropout

Figure 3.8: Dropout neural network: (a) Original model with 1 hidden layer (b) Dropout model with a drop rate of 0.5

Srivastava et al. explored the effectiveness of dropout using MNIST dataset for image classification tasks and CIFAR-10 dataset for object recognition tasks, the result indicated that random dropout could effectively reduce the error for both benchmark tasks, from 1.60% to 1.35% and from 14.98% to 12.61% respectively.

3.3.6 Data Augmentation

Data augmentation is a practical technique to artificially expand the dataset, not only increasing the amount but also promoting the diversity of the dataset. Due to reasons such as privacy protection for patients and deep expertise in clinical applications, large scale medical image data with good-quality annotations is not easily accessed. However, the performance of deep learning models heavily depends on the amount and quality of training image dataset to improve the generalization ability. The more diverse the training dataset is, the more robust the deep learning model will be in segmentation tasks for unseen image data. In one of the early applications of CNNs, LeNet-5 [106], the dataset of handwritten digits was

warped for better image classification performance; the AlexNet proposed by Krizhevsky et al. [107] used randomly cropped and flipped image augmentation techniques, which reduced overfitting in deep neural networks.

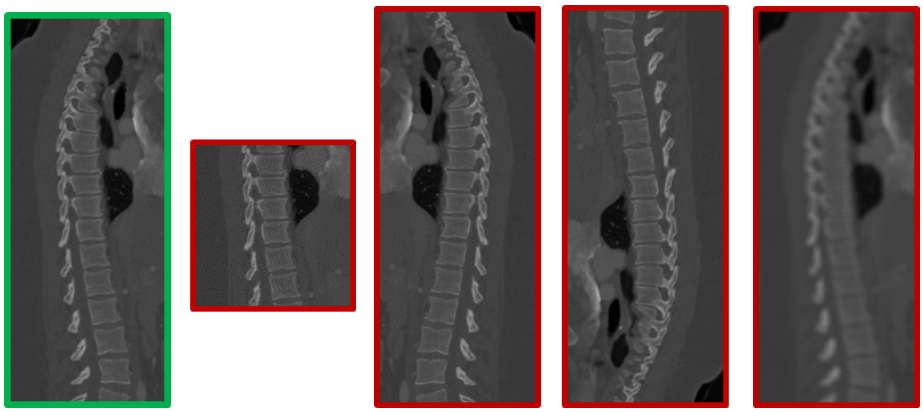


Figure 3.9: Data augmentation of dataset: (a) Original image (b) Randomly cropped image (c) Flipped image (left to right) (d) Flipped image (upsidedown) (e) Gaussian blurred image

In this research, several typical data augmentation methods were applied, as shown in Fig. 3.9, including randomly cropping the image data in three dimensions, flipping the image from left to right or upside down, and adding Gaussian blur on the image with a standard deviation of 1 for the Gaussian kernel. The principle of designing these strategies is to create various image datasets under different situations while keeping the main features unchanged.

During training, both the original CT image data and the corresponding label will apply the above data augmentation techniques each with a probability of 50%.

3.4 Experiment Results

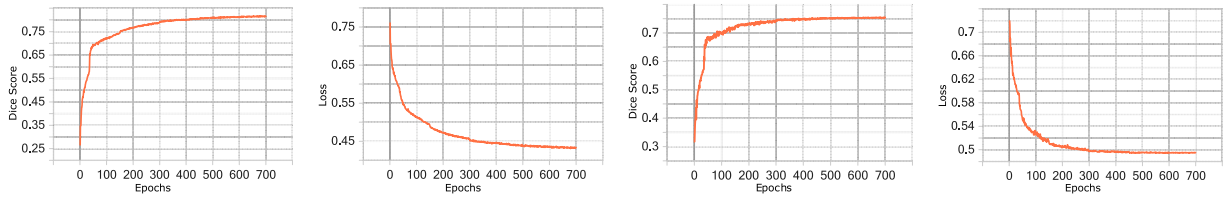
The four neural networks in Section 3.1 were implemented using *PyTorch 1.10 (Paszke et al.)* [108], an open-source Python [109] package of machine learning framework. The total

number of training epochs was 700 and the batch size was 1 due to memory limitation, namely in every epoch all the training data will be propagated through the network one after another. The four deep learning models were trained on a GTX 1080 GPU with 8 GB memory. The initial learning rate was 0.0001 and would be divided by 2 after every 150 epochs; in each epoch, the learning rate was adaptively adjusted according to the Adam optimizer. The initial coefficient of auxiliary loss for the deep supervision mechanism was 0.4 and would be multiplied by 0.8 after every 30 epochs. The early stopping was applied during training: if the validation Dice score does not improve after 100 epochs, the training progress will be forced to stop. A dropout rate of 0.2 was used in ResUNet. Other training strategies were used as per Section 3.3.

Table 3.1 shows the number of parameters of each model, results of training and validation Dice score, and total training epochs and time. The number of parameters goes up in direct proportional to the complexity of the model. The ResUNet achieved the best validation Dice score of 0.9420 and the KiUNet achieved the best training Dice score of 0.9506. It is worth noting that although the number of parameters of KiUNet is quite less than the one of ResUNet, the two models had similar training time due to high memory usage of KiUNet. Fig. 3.10 $\sim$ Fig. 3.13 showed the learning curves of four neural networks including the Dice score and the loss curves in both training and validation stages. The validation Dice score and validation loss data were collected after every training epoch. With the decline of the loss over time, the four models successfully converged and the Dice score had an overall growth despite some oscillations. Fig. 3.14 shows the segmentation results of one CT slice from the validation dataset during training of the SegNet at 10, 20, 30, 40, 50, 60, 70, 80, 90, 150, 300, 600 epochs.

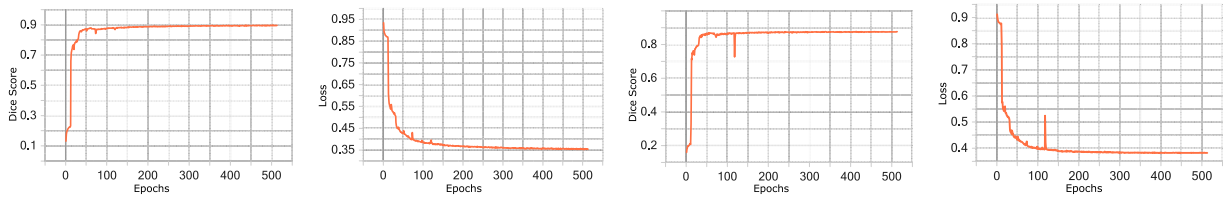
Table 3.1: Training details and segmentation results of SegNet, UNet, ResUNet and KiUNet.

	SegNet	UNet	ResUNet	KiUNet
Number of parameters	2326190	3939844	9498744	2329264
Training Dice Score	0.8177	0.8979	0.9473	0.9506
Validation Dice Score	0.7549	0.8772	0.9420	0.9311
Training Time	1d13h	1d7h	1d23h	1d23h
Epochs	700	513	591	569



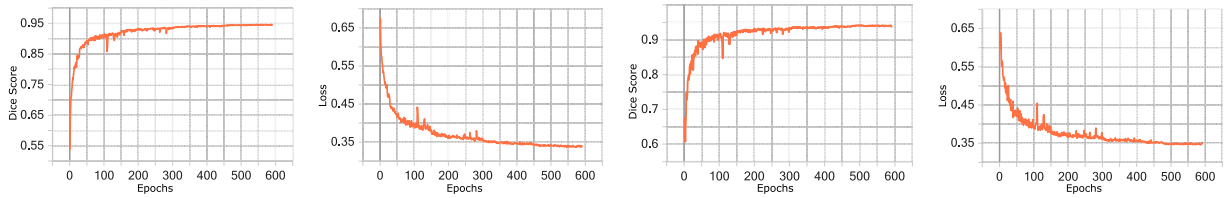
(a) **(b)** **(c)** **(d)**

Figure 3.10: Learning curves of SegNet: (a) Training Dice Score (b) Training Loss (c) Validation Dice Score (d) Validation Loss



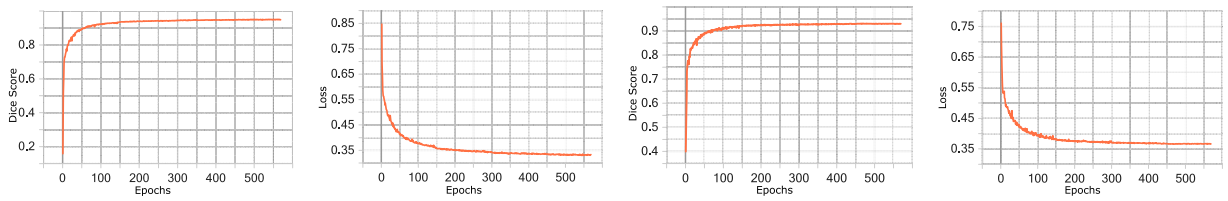
(a) **(b)** **(c)** **(d)**

Figure 3.11: Learning curves of UNet: (a) Training Dice Score (b) Training Loss (c) Validation Dice Score (d) Validation Loss



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

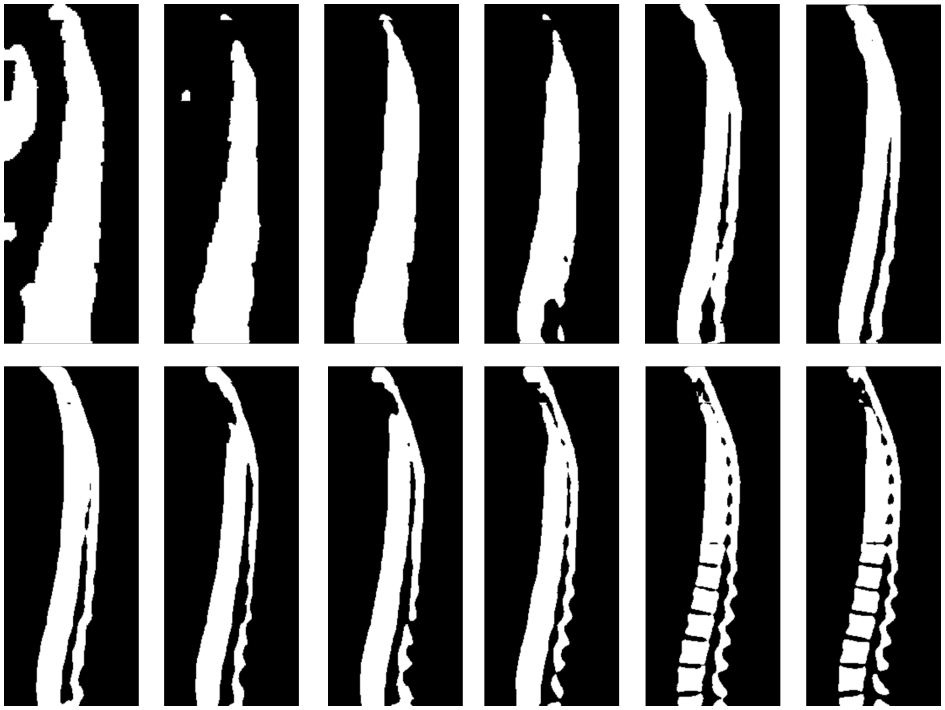
Figure 3.12: Learning curves of ResUNet: (a) Training Dice Score (b) Training Loss (c) Validation Dice Score (d) Validation Loss



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

Figure 3.13: Learning curves of KiUNet: (a) Training Dice Score (b) Training Loss (c) Validation Dice Score (d) Validation Loss

(a)



(b)

Figure 3.14: Segmentation results of SegNet during training at 10, 20, 30, 40, 50, 60, 70, 80, 90, 150, 300, 600 epochs. (a) Axial plane (b) Sagittal plane

3.5 Reconstruction of the Dried-out Vertebra

To test the performance of the four neural networks in clinical applications, a human excised dried-out lumbar vertebra was introduced. The dried-out vertebra was scanned using computed tomography (CT) at *McGill University Health Centre*. The corresponding CT image data has a dimension of 972×972×1247 with the image intensity from -999.7 to 8233. Both the slice thickness and pixel spacing are 0.08508 *mm*.

To keep the format of the input image data consistent with the training data, the CT image intensity of the dried-out vertebra was thresholded between -2500 and 3500, and the voxel spacing was resampled to 1×1×1 *mm*. To maintain the real scale of the vertebra, the image size was also reset to 83×83×107 according to Equation 3.2.

The preprocessed CT image data of the dried-out vertebra was then input into four neural networks using the best training models respectively to obtain the four different masks, which is shown in the red area of Fig.3.15. The CT image of the dried-out vertebra was also manually labelled using an open-source software *ITK-SNAP (University of Pennsylvania, PA)* [110]. The Dice scores of the automatically segmented masks of the dried-out vertebra compared to the manual label were 0.9002, 0.9333, 0.9483 and 0.9715 respectively for the four networks. The Marching Cubes algorithm proposed by Lorensen et al. [50] was used to reconstruct the model according to the above five masks. The reconstructed models were shown in yellow in Fig.3.15.

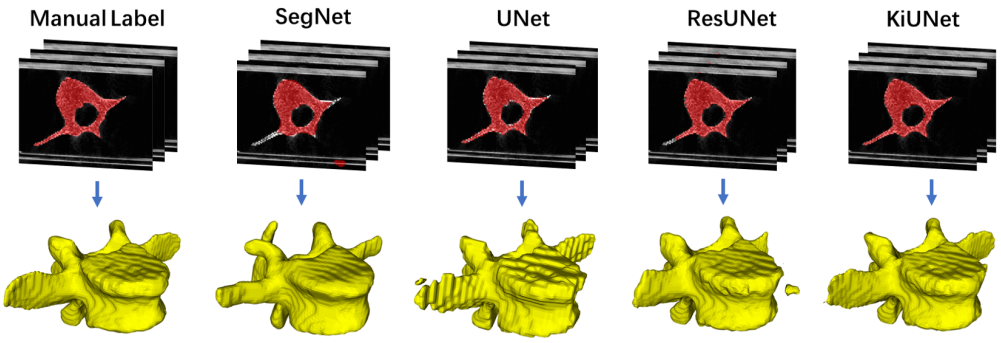


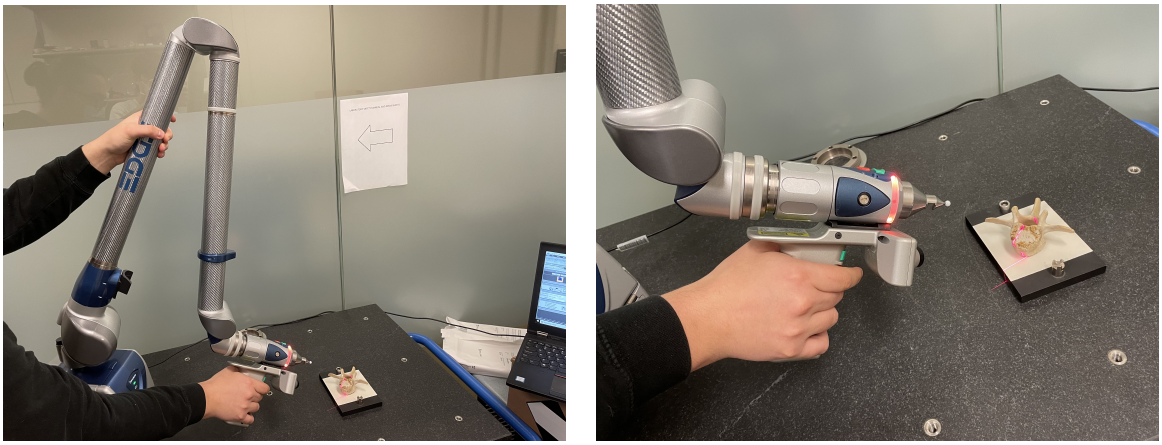
Figure 3.15: Reconstructed models of the dried-out vertebra according to different masks.

3.6 Reverse Engineering of the Dried-out Vertebra

In this section, an accurate reverse engineering solution using the *FARO Arm*, a high-precision measuring robotic arm, is proposed to obtain the original geometry of the dried-out vertebra, which will serve as the gold standard of the reconstructed virtual model. Several post-processing procedures of the point clouds after scanning were conducted to restore the geometry of the vertebra as much as possible.

3.6.1 FARO Arm Scanning

To obtain the original geometry of the dried-out vertebra with high accuracy, the *FARO Edge ScanArm HD (FARO Technologies, Lake Mary, FL)* was used to capture the point clouds of the model. The *FARO Edge ScanArm HD* is a 7-axis, articulated arm with a spherical working volume. A laser line probe can be attached to the end effector of the robotic arm for three-dimensional scanning. According to the technical specifications [111], the robotic arm has an accuracy of $\pm 25\ \mu m$ and repeatability of $25\ \mu m$, 2σ, and the maximum scan rate of the laser probe can reach 560,000 points/sec.



(a) **(b)**

Figure 3.16: The *FARO Arm* with laser line probe attached to the end effector for 3D scanning.

The *Geomagic Studio 2014*[1] was used for communication and parameters setting between the *Faro Arm* and the computer. It was also used for post-processing procedures of the raw data in the following steps.

A plane compensation of the laser probe was first applied to look for reference of the plane on which the vertebra was ready for scanning. Then, during scanning, the spacing of the ordered data was set to 0.013 mm. The dried-out vertebra was first scanned in the upper half and was then flipped to scan the bottom half to ensure there was a complete vertebra included in the point clouds, as shown in Fig. 3.16.

3.6.2 Point Clouds Post-processing

After the two sets of point clouds were collected, several post-processing procedures were conducted to transform the raw 3D scanned data into a complete vertebra as surface

[1]A Toolbox for transforming 3D scanned data into surface and native CAD models, developed by 3D Systems, inc.

mesh.

The point clouds of the table, which were also scanned in as part of the raw data, were firstly removed. Secondly, as the two sets of point clouds, the upper and bottom halves of the vertebra, both include the same transverse process, a rough registration using landmarks was applied. Four landmarks were selected on the tranverse process of the two sets of point clouds in the same sequence, as shown in Fig. 3.17.

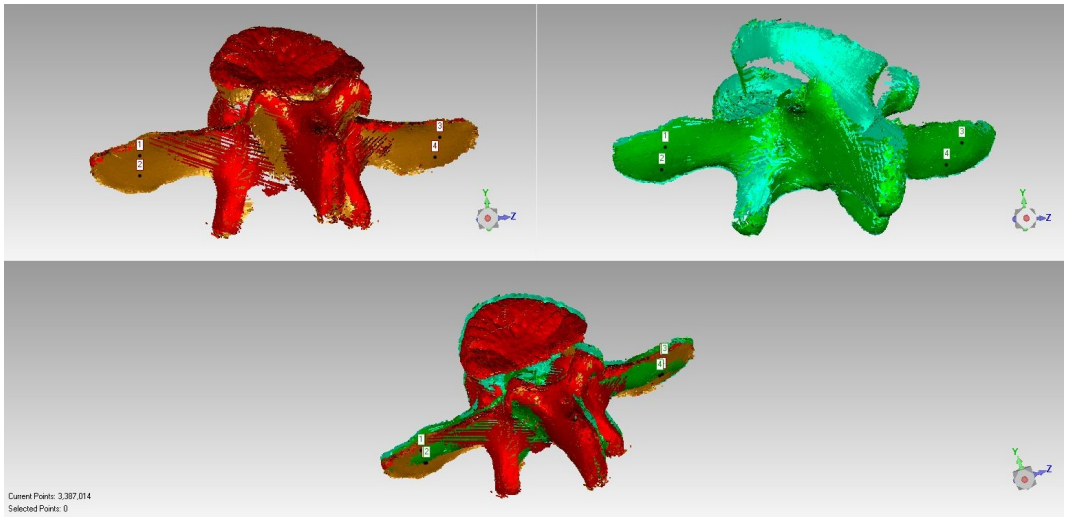


Figure 3.17: A rough registration based on four landmarks selected on the transverse process for each half of the vertebra.

A global registration of two halves of the vertebra was then conducted using functions in *Geomagic Studio* with a result of an average distance of 0.1056 *mm* and standard deviation of 0.1418 *mm* after 18 iterations, as shown in Fig. 3.18.

The point clouds of the complete vertebra were then transformed into surface mesh. Fig. 3.19 shows the final result of the scanned dried-out vertebra composed of very high accurate surface mesh with 3,314,916 triangles.

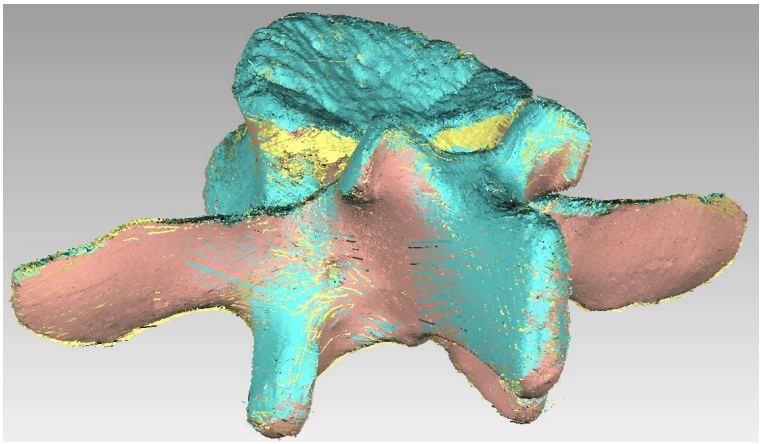


Figure 3.18: Global registration based on the two halves of the vertebra.

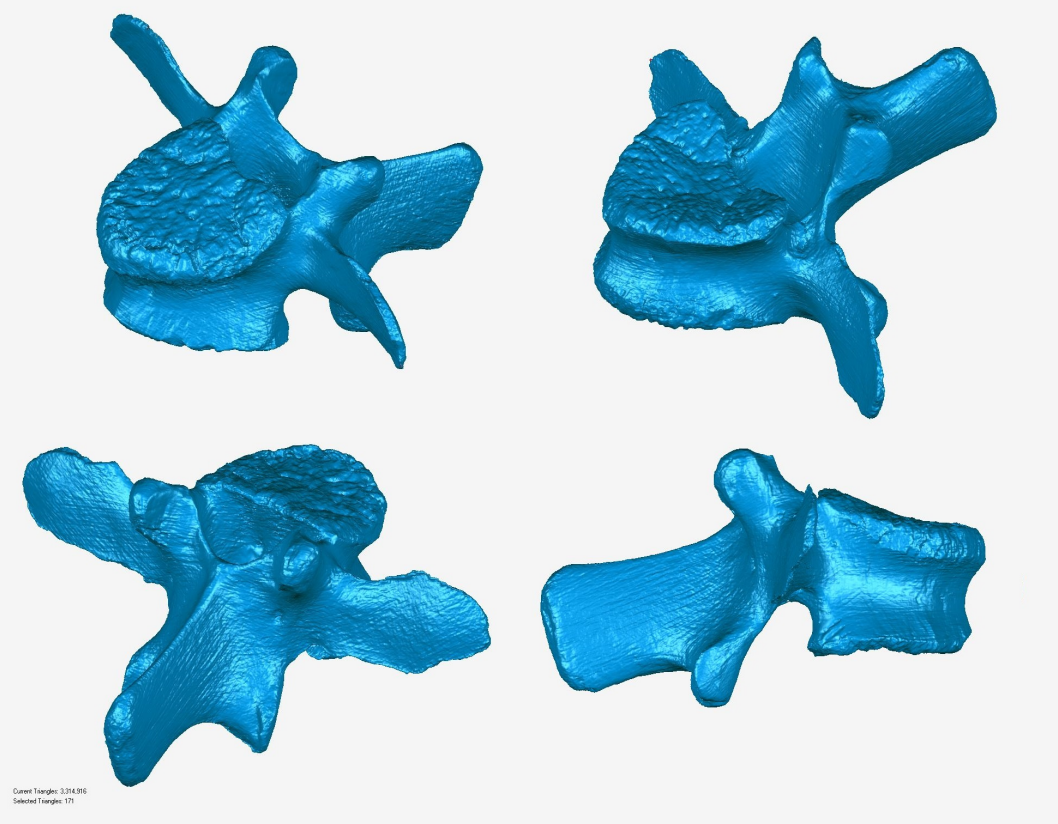


Figure 3.19: Final result of the scanned dried-out vertebra.

Chapter 4

Model Evaluation of Reconstructed Vertebra

In this chapter, to evaluate the accuracy of the reconstructed models compared with the gold standard model, several 3D model evaluation metrics were proposed as well as the heatmap was used for error visualization and the histogram to show error distribution.

4.1 Metrics for Three-dimensional Model Evaluation

In this section, several evaluation metrics originally designed for two-dimensional image segmentation tasks including Dice score, IoU and Hausdorff distance, were modified for 3D surface mesh evaluation. The heatmap and histogram as well as some basic 3D model evaluation metrics were also introduced.

4.1.1 Sørensen - Dice Coefficient

The Sørensen - Dice coefficient, which is also known as the Dice similarity coefficient (DSC) or F1 score, was first proposed by Lee R. Dice [66] in 1945 to gauge the similarity of two species. The original formula was applied to discrete data, as shown in Equation 4.1.

$$DSC = 2\frac{S_A \bigcap S_B}{S_A + S_B} \tag{4.1}$$

The Dice coefficient was first introduced to the field of medical image segmentation by Zijdenbos et al. [89] in 1994 for comparing the difference between pixel-based images. Since then, the Dice coefficient, as well as many of its variations [112], has become one of the essential metrics to compare the labeled masks against the gold standard masks. As the image segmentation task belongs to the binary classification, the Dice coefficient can also be denoted using the confusion matrices of true positive (TP), false positive (FP), and false negative (FN), which stands for pixels correctly segmented as foreground, pixels falsely segmented as foreground and pixels falsely detected as background respectively.

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

Different from the pixel-wise accuracy, the Dice coefficient can not only reflect the accuracy but also penalize the false positive data and focus more on perceptual quality. Due to this feature, it was later adapted to a loss function for semantic segmentation tasks in deep learning by Carole H Sudre et al. [90] in 2017, known as the Dice Loss:

$$Dice\ Loss = 1 - 2\frac{S_A \bigcap S_B}{S_A + S_B} \tag{4.3}$$

However, as the Dice coefficient is generally used to compare the similarity between two-dimensional images, it was modified in this section for 3D model evaluation. In Equation 4.4, the modified Dice coefficient is twice the intersection over the sum of two volumes A and B. The result ranges between 0 and 1. And when the result equals 1, the two models are considered the same one.

$$3D\ DSC = 2\frac{V_A \bigcap V_B}{V_A + V_B} \in [0, 1] \tag{4.4}$$

4.1.2 Intersection over Union

The intersection over union (IoU), also known as the Jaccard similarity coefficient or Jaccard index, is another commonly used evaluation metric for medical image segmentation. The IoU was first designed by Paul Jaccard [67] in 1912 to measure the similarity between finite sample sets. It was later broadly used in semantic segmentation for its outstanding evaluation performance.

As its name suggests, the IoU computes the ratio of the overlapped area between the predicted mask and the ground truth mask to the union area of the predicted mask and the ground truth mask. Equation 4.5 indicates that the IOU always ranges from 0 to 1, signifying a poor segmentation result to a perfect segmentation result.

$$IoU = \frac{S_A \bigcap S_B}{S_A \bigcup S_B} \tag{4.5}$$

Similar to the Dice coefficient, the IoU can reflect both the value and localization of the masks. Actually, the IoU and Dice coefficient are monotonically related: $IoU = Dice/(2 - Dice)$. In image segmentation tasks, the IoU can be described using confusion matrices for binary classification, as shown in Equation 4.6.

$$IoU = \frac{TP}{FP + TP + FN} \tag{4.6}$$

The IoU can also serve as the geometrical-based loss function when training deep neural networks for image segmentation. It was first used in 2009 by Polak et al. [113] and has a more common name of Jaccard loss.

In this section, the formula of IoU was extended to fit with three-dimensional models. In Equation 4.7, the modified IoU is the intersection of two volumes A and B over the union of two volumes A and B. The result ranges between 0 and 1. And when the result is equal to 1, the two models are considered the same one.

$$3D\ IoU = \frac{V_A \bigcap V_B}{V_A \bigcup V_B} \in [0, 1] \tag{4.7}$$

4.1.3 Hausdorff Distance

The Hausdorff distance (HD) was first proposed by Felix Hausdorff in 1914 to measure the degree of mismatch between two subsets of a metric space in his book *Grundzüge der Mengenlehre*. During past decades, Hausdorff distance had many applications in computer vision such as object matching [114], image comparison [115] and measuring surface errors [116], etc.

Given two non-empty point sets $A = \{a_1, a_2, ..., a_m\}$ and $B = \{b_1, b_2, ..., b_n\}$, the shortest distance of point a to set B is defined as:

$$d(a, B) = \min_{b \in B} \|a - b\| \tag{4.8}$$

where $\|\cdot\|$ denotes the L^2 norm. Then, the one-sided Hausdorff distance from A to B is defined as:

$$h(A, B) = \max_{a \in A} d(a, B) \tag{4.9}$$

Equation 4.9 aims to find the farthest distance among the shortest distances of every point in A from all points in B. The Hausdorff distance from B to A is defined as $h(B, A)$ in the same way. The function of $h(A, B)$ and $h(B, A)$ is asymmetric, which in general they will have different results. Therefore, a bi-directional Hausdorff distance between A and B is defined as:

$$H(A, B) = \max\{h(A, B), h(B, A)\} \tag{4.10}$$

Different from metrics such as the Dice coefficient and IoU index which are composed of the interior area of the mask, the Hausdorff distance is very sensitive to the contour of the segmented masks. Even a single outlying point will greatly affect the result of Hausdorff distance of that point. Hence, the 95^{th} percentile of Hausdorff distances between boundary points in A and B (95% Hausdorff distance) is frequently used to eliminate the impact of those small subsets of outliers.

To measure the differences between two 3D surfaces in triangular meshes A and B using Hausdorff distances, a common solution is to take each vertex a from mesh A to search for the closest vertex b on mesh B. The farthest distance among all those shortest distances of vertexes between mesh A and B is the one-sided Hausdorff distance. And then, vice versa to compute the other-sided Hausdorff distance. To improve the accuracy and to reduce memory usage and unnecessary computations of the above method, Aspert et al. [116] appropriately implemented the Hausdorff distance algorithm by adding more sample points on each triangular mesh and adding the points to triangle distance evaluation.

In practice, to clearly show the different areas and range of errors, the colorful heatmap is often used to compare the surface quality. Each Hausdorff distance of the sample points on the surface mesh was computed and stored so that the range of error can be colored on each triangular mesh until covering its whole surface.

4.1.4 Basic Evaluation Metrics

Assume there are N sample points on mesh A. Based on Equation 4.8, we can compute each shortest distance d_i of all sample points on mesh A as Equation 4.11:

$$d_i(a, B) = \min_{b \in B} \|a - b\| \tag{4.11}$$

Using every d_i, the following useful metrics can be also modified to further evaluate the 3D surface mesh:

Maximum Sample Error, which quantifies the maximum error between two surface meshes:

$$Max\ Sample\ Error = Max(d_i) \tag{4.12}$$

Minimum Sample Error, which quantifies the minimum error between two surface meshes:

$$Min\ Sample\ Error = Min(d_i) \tag{4.13}$$

Mean Sample Error, which quantifies the mean error between two surface meshes:

$$MSE = \bar{d} = \frac{1}{N}\Sigma_i d_i \tag{4.14}$$

Standard Deviation Error, which quantifies the standard deviation of the sample error:

$$STD = \sqrt{\frac{1}{N-1}\Sigma_i (d_i - \bar{d})^2} \tag{4.15}$$

Mean Square Distance, which computes the mean square distance of all sample errors:

$$MSD = \frac{1}{N}\Sigma_i d_i^2 \tag{4.16}$$

Mean Absolute Distance, which computes the mean absolute distance of all sample errors:

$$MAD = \frac{1}{N}\Sigma_i |d_i| \tag{4.17}$$

It is worth noting that the above metrics are all one-directional.

4.2 Model Registration

To compare the difference between two models, model registration is necessary as the *FARO Arm* scanning coordinate system of the gold standard model is different from the CT coordinate system of the reconstructed vertebra. A homogeneous transformation matrix was used to apply the rigid body transformation to the gold standard model.

To obtain the transformation matrix, a coarse registration based on landmarks was first introduced. Two sets of landmarks were selected in the same order on approximate same positions of the surface mesh of the two different models, as is shown in Fig.4.1a. Assume the two point sets, P and Q, each having n ordered points, $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_n)$, the landmark registration is designed to find the matrix $\widehat{T}$:

$$\widehat{T} = \underset{T}{\arg\min} \sum_{i=1}^{n} |p_i - q_i|^2 \tag{4.18}$$

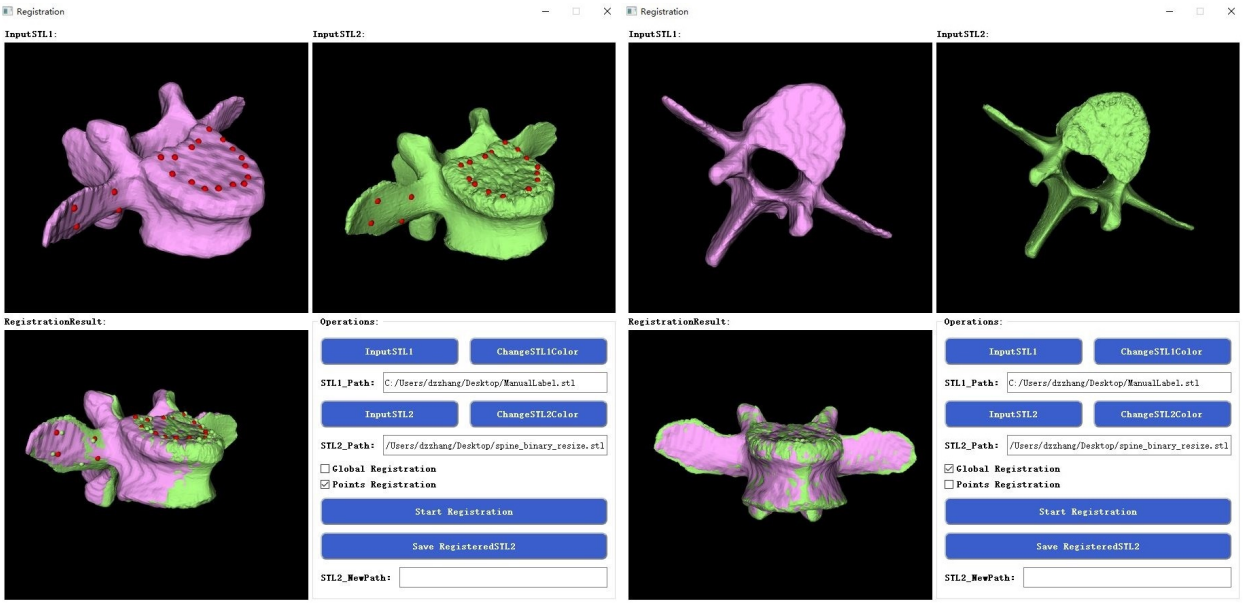
Based on the matrix $\widehat{T}$, the Iterative Closest Point (ICP) algorithm was then applied as a more accurate global registration. The ICP algorithm was first proposed independently by Chen et al. [117] and Besl and McKay [118], which is capable of estimating a transformation T that best aligns two point sets $P = (p_1, p_2, ..., p_n)$ and $Q = (q_1, q_2, ..., q_m)$, where in general $n \neq m$. In this section, P and Q are the vertices of the two surface meshes. In the first step, given the initial rotation matrix R and translation matrix t from $\widehat{T}$, the source point sets will be transformed to a new position and orientation. The second step is to compute the nearest point in the set P for every point in the set Q using Euclidean distance without requiring the correspondences between P and Q:

$$d_i = min\{\sqrt{p_n^2 - q_m^2}\} \tag{4.19}$$

The outliers that have a distance d_i larger than a threshold will be removed. In the third step, the homogeneous transformation matrix T including rotation matrix R and translation matrix t is calculated using Singular Value Decomposition (SVD) to minimize the error:

$$E(R,t) = \sum_{i=1}^{m}\sum_{j=1}^{n}\omega_{ij}\|q_i - (Rp_j + t)\|^2 \tag{4.20}$$

where $\omega_{ij} = 1$ if point q_i is the nearest point to p_j. If the error is higher than the given threshold, the R and t will be input into the first step and all the three steps will be iterated; otherwise, the T is the optimal transformation matrix.



(a) Landmark Registration **(b)** ICP Registration

Figure 4.1: A practical GUI software for model registration.

To realize the functions of selecting points on the surface mesh, a practical software[1]

[1]https://github.com/dzzhang96/Points-Registration-ICP

with graphical user interface (GUI) was developed for this research based on *Visualization Toolkit (VTK) 7.0 (Kitware Inc.)* [119], an open-source library for 3D computer graphics and *Qt 5.14.2 (The Qt Company, Espoo, Finland)*, a widget toolkit for creating GUI, as shown in Fig.4.1.

By right-clicking on the surface mush, the software can render a red sphere on the clicked point and collect the corresponding coordinates. After two sets of points were selected, the landmark registration and the ICP registration using two *VTK* modules, the vtkLandmarkTransform class[1] and vtkIterativeClosestPointTransform class[2] were conducted to obtain the homogeneous transformation matrix T, which was then applied on the target model to complete the model registration. The maximum iteration number was set to 50 for the ICP registration.

4.3 Experiment Results

In this section, the accuracy of five different dried-out vertebra models reconstructed from output masks of the manual label, SegNet, UNet, ResUNet and KiUNet was evaluated using metrics in Section 4.1 as well as the heatmap for error visualization and the histogram for error distribution.

The evaluation methods were calculated based on the structure of *Meshvalmet 3.0 (University of North Carolina at Chapel Hill, NC)*, an open-source tool measuring surface to surface distance between two triangle meshes using *VTK* library [119] for computation and visualization. *Meshvalmet 3.0* implemented the methods of Nicolas Aspert et al. [116]

[1]https://vtk.org/doc/nightly/html/classvtkLandmarkTransform.html
[2]https://vtk.org/doc/nightly/html/classvtkIterativeClosestPointTransform.html

to efficiently select the sample points and estimate the distance between discrete triangular 3D meshes with adjustable user-specified uniform sampling level.

4.3.1 Different Metrics Results

Before evaluation, to keep the number of triangles of the reconstructed models and the *FARO Arm*-scanned model in the same order of magnitude, the latter was reduced to 57,332 triangles while maintaining the original geometry as much as possible. Due to over-segmentation volumes in the reconstructed models, those outliers that were not attached to the vertebra were removed so that the evaluation results can better focus on the accuracy of the main body of the vertebra.

As most of the metrics in Section 4.1 are one-directional, the results will be different when comparing the reconstructed model with the gold standard model, and in turn, when comparing the gold standard model with the reconstructed model. Therefore, the evaluation will include both two-directional results.

Table 4.1 shows the evaluation results of the model reconstructed from the manual label. The reconstructed model achieved a Dice score of 96.41%, IOU of 93.08% compared with the gold standard model and the maximum Hausdorff distance was 2.27 *mm*. The standard deviation error of all sample points, the mean square distance and mean absolute distance were both low in two-directional results, indicating that the model was quite close to the gold standard geometry. However, the drawback is that the manual label took around an hour to label on the CT image data and required examination by professional radiologists, which is time-consuming in clinical applications. And this model still shows minor errors

compared to the *FARO Arm*-scanned vertebra.

Table 4.1: Evaluation results of the vertebra reconstructed from the manual label.

A: Gold Standard	**Vertices: 28640**		**Triangles: 57332**	
B: Manual Label	**Vertices: 17476**		**Triangles: 34952**	
Sampling Steps	**Min Sampling Frequency**		**Number of Bins**	
0.5	2		256	
A→B				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
2.2666	0.964149	0.93078	0.106805	0.246156
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
-0.0908982	0.313915	-2.2666	2.112402	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.246156	0.21497	0	2.2666	
B→A				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
1.188989	0.964149	0.93078	0.062877	0.200327
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.137428	0.209741	-0.92042	1.188989	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.200327	0.150819	0	1.188989	

Table 4.2 shows the evaluation results of the model reconstructed from the output mask of SegNet. The reconstructed model has an over-segmentation volume on the spinous process, which can be seen in Fig. 4.2 so that the maximum Hausdorff distance (or the maximum sample error) came to 20.80 *mm* and the mean square distance was 7.11 *mm*. The model achieved a Dice score of 88.29%, IOU of 79.03%. When the reconstructed model was overlaid on the gold standard model, the heatmap in Section 4.3.2 shows a large blue area, which means the model is actually smaller than the gold standard model.

Table 4.3 shows the evaluation results of the model reconstructed from the output mask of UNet. The reconstructed model achieved a Dice score of 92.43%, IOU of 85.93% and the maximum Hausdorff distance was 3.24 *mm*. The standard deviation error of sample points, the mean square distance and mean absolute distance were low in both two-directional

Table 4.2: Evaluation results of the vertebra reconstructed from SegNet mask.

A: Gold Standard		**Vertices: 28640**	**Triangles: 57332**	
B: SegNet		**Vertices: 16600**	**Triangles: 33200**	
Sampling Steps		**Min Sampling Frequency**	**Number of Bins**	
0.5		2	256	
A→B				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
7.59992	0.882875	0.79031	1.60269	0.863267
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.181716	1.25287	-4.51087	7.59992	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.863267	0.925994	0	7.59992	
B→A				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
20.8031	0.882875	0.79031	7.10579	1.06772
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.6965	2.57308	-2.39903	20.8031	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
1.06772	2.4425	0	20.8031	

evaluations. However, several stripes on the surface of the model can be observed in Fig. 4.2, which cannot be figured out from the results in table 4.3. In this case, the histogram in Fig. 4.3 was used as a practical method to present the distribution of bins of sample points at different Hausdorff distances. When the reconstructed model was overlaid on the gold standard model, the curve shows distinct oscillations in both diagrams of signed distance and absolute distance.

Table 4.4 shows the evaluation results of the model reconstructed from the output mask of ResUNet. Although the ResUNet achieved excellent results in the validation dataset, it failed to segment the transverse process of the dried-out vertebra (Fig. 4.2) so that the reconstructed model had a Dice score of 91.41%, IOU of 84.18%, even lower than the results of the UNet model. This may happen in all UNet variations due to its undercomplete architecture. When the ResUNet network comes deeper, it will more likely focus on high-

Table 4.3: Evaluation results of the vertebra reconstructed from UNet mask.

A: Gold Standard		**Vertices: 28640**		**Triangles: 57332**
B: UNet		**Vertices: 17082**		**Triangles: 34164**
Sampling Steps		**Min Sampling Frequency**		**Number of Bins**
0.5		2		256
A→B				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
3.240766	0.924302	0.859258	0.398099	0.493476
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
-0.043914	0.629423	-2.917733	3.240766	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.493476	0.393168	0	3.24066	
B→A				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
1.601908	0.924302	0.859258	0.298134	0.443916
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.189396	0.512118	-1.472695	1.601908	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.443916	0.31792	0	1.601908	

level features to extract large structure annotations but ignore structures of boundaries. On the other hand, the main body of the ResUNet reconstructed vertebra still has very high accuracy. As is shown in table 4.4, when the reconstructed model was compared to the gold standard model, the mean square distance was only 0.2698 *mm* and the mean absolute distance was 0.4113 *mm*.

Table 4.5 shows the evaluation results of the model reconstructed from the output mask of KiUNet. The KiUNet output mask, integrating both the advantages of KiNet and UNet, performs the best in all the automatically reconstructed models. The reconstructed model achieved a Dice score of 92.79%, IOU of 86.56% and the maximum Hausdorff distance was 3.85 *mm*. Compared with the gold standard model, the standard deviation error of sample points was 0.2815 *mm*, the mean square distance was 0.2459 *mm* and the mean absolute distance was 0.4083 *mm*.

Table 4.4: Evaluation results of the vertebra reconstructed from ResUNet mask.

A: Gold Standard		**Vertices: 28640**	**Triangles: 57332**	
B: ResUNet		**Vertices: 15304**	**Triangles: 30608**	
Sampling Steps		**Min Sampling Frequency**	**Number of Bins**	
0.5		2	256	
A→B				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
19.153991	0.91413	0.841841	7.584956	1.159835
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.728901	2.65588	-5.607024	19.153991	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
1.159822	2.497935	0	19.153991	
B→A				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
3.273735	0.91413	0.841841	0.269847	0.411342
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.109184	0.507866	-3.273735	1.435534	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.411342	0.317247	0	3.273735	

Table 4.5: Evaluation results of the vertebra reconstructed from KiUNet mask.

A: Gold Standard		**Vertices: 28640**	**Triangles: 57332**	
B: KiUNet		**Vertices: 16988**	**Triangles: 33972**	
Sampling Steps		**Min Sampling Frequency**	**Number of Bins**	
0.5		2	256	
A→B				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
3.85562	0.927957	0.865597	0.388774	0.482925
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
-0.0165427	0.6233	-3.01027	3.85562	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.482925	0.394409	0	3.85562	
B→A				
HD (*mm*)	**Dice Index**	**IOU**	**MSD (*mm*)**	**MAD (*mm*)**
2.11992	0.927957	0.865597	0.245989	0.408279
Signed Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.149168	0.473012	-2.11992	1.6358	
Absolute Distance				
MSE (*mm*)	**STD (*mm*)**	**Min Sample Error (*mm*)**	**Max Sample Error (*mm*)**	
0.408279	0.281599	0	2.11992	

4.3.2 Heatmap

Figure 4.2 is the heatmap visualizing the differences between two vertebra models. The Hausdorff distance of every sample point was computed. In this study, a blue-green-red map was used, where blue stands for negative errors and red means positive errors. Two different types of distance were applied: the absolute distance converts all the errors to positive values while the signed distance shows the actual values of the error. The heatmaps in Fig.4.2 also include two-directional results, the reconstructed model being mapped on the gold standard model, and in turn, the gold standard model being mapped on the reconstructed model.

4.3.3 Histogram

Figure 4.3 is the histogram showing the distribution of bins of sample points error at different Hausdorff distances. The number of bins was 256 in all cases. Both the signed distance and the absolute distance were used to observe the error distribution from different aspects. The oscillations of the curve can reflect the surface quality compared with the gold standard model such as the stripes on the UNet model.

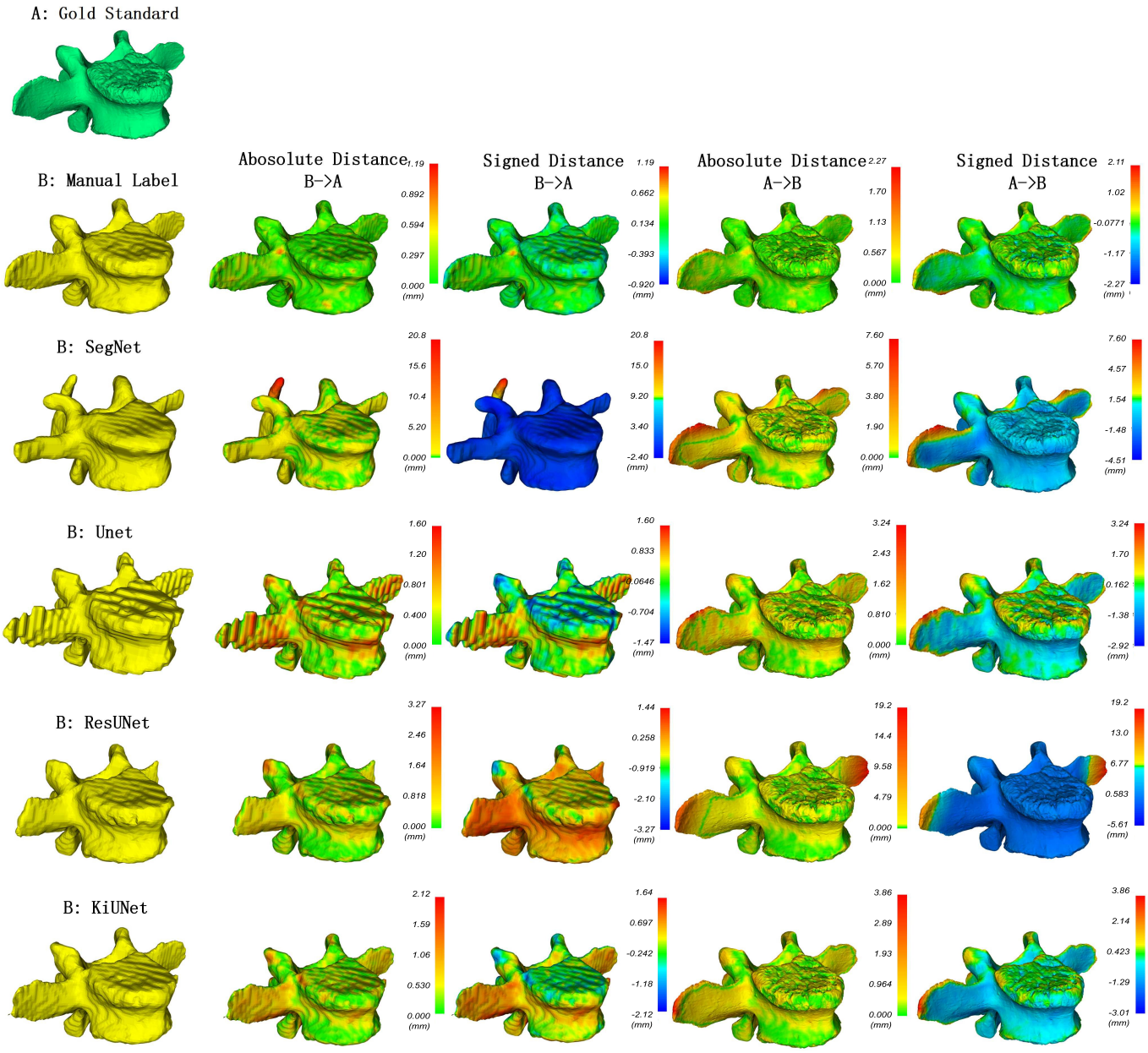


Figure 4.2: Heatmaps for error visulization between gold standard vertebra and reconstructed vertebra.The results include both two directions of Hausdorff distance.

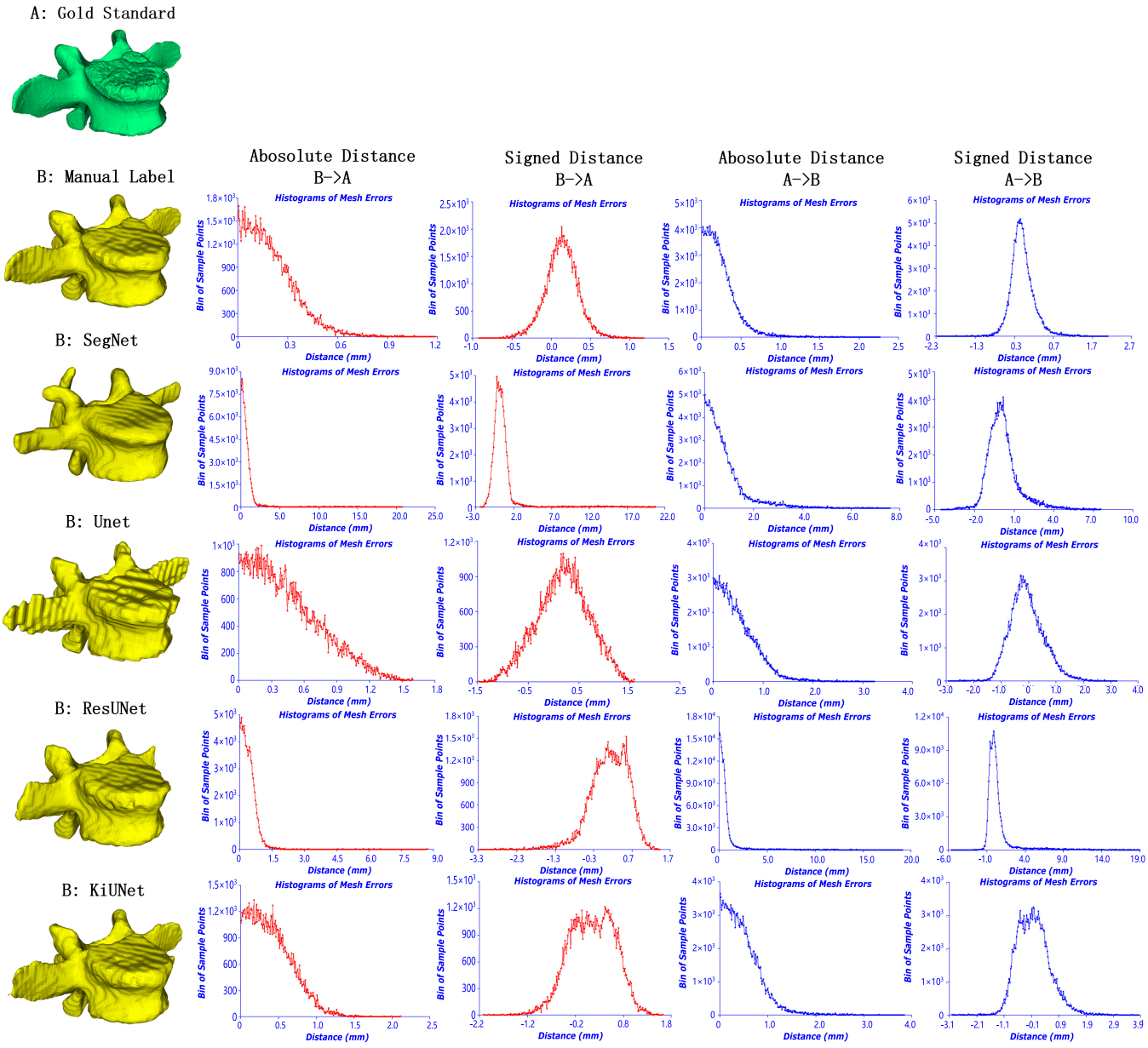


Figure 4.3: Histograms showing error distribution of bin of sample points at different Hausdorff distances.

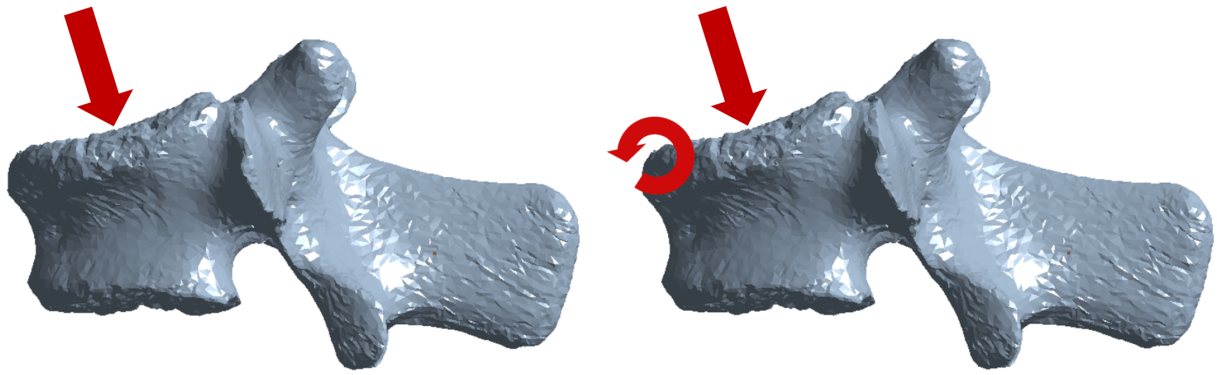
Chapter 5

Finite Element Analysis

Finite element analysis (FEA) is a practical tool for numerical simulation. Nowadays, many applications of FEA have been used in numerous medical researches, providing a platform for surgeons and researchers to accurately simulate the biomechanics for both healthy and pathological situations. Due to the complex structure and biomechanics of the spine, relevant study of finite element modelling in spine has been conducted as early as 1957 [120].

In this research, an evaluation method using FEA was used to explore the impact of different geometries of the reconstructed vertebra on biomechanical results. In the meantime, the FEA can evaluate the critical partial surface accuracy of the vertebra from another perspective. Because even minor errors in the surface area between the intervertebral disc and the vertebra will be magnified under spinal loads and hence greatly affect the biomechanical results.

To obtain effective FEA results, two experiments were designed considering the biomechanics of human lumbar spine and were conducted on all the reconstructed



(a) Follower Load **(b)** Follower Load + Bending Moment

Figure 5.1: Two FEA experiments on the dried-out vertebra models with (a) Follower load of 1000 N (b) Follower load of 1000 N + Bending moment of 7.5 $N \cdot m$. The follower load was added vertically to the top surface area of around 6.8 $\times 10^{-4}$ m^2 while the bending moment was applied to the whole body of the vertebra. The bottom surface area of the vertebra served as the fixed support.

dried-out vertebra models and the *FARO Arm*-scanned gold standard model. Experiment 1 applied the follower load on the top surface area of the vertebra, covering an area of around 6.8 $\times 10^{-4}$ m^2. And the bottom surface of the area served as the fixed support. The follower load [121] was first proposed by Patwardhan et al. to define a compressive vertical load whose direction is along the tangent to the curve of the lumbar spine. Based on Experiment 1, Experiment 2 added a pure bending moment to simulate the flexion/extension. Fig. 5.1 shows the two experiments on the dried-out vertebra.

The follower load was 1000 N which approximates the compression on the lumbar spine when standing and walking [121, 122]. The bending moment was set to 7.5 $N \cdot m$ [49, 123]. The material properties of the dried-out vertebra body were assumed isotropic elasticity with the Young's modulus of 5 $\times 10^9 Pa$ and the Poisson's ratio of 0.3 according to previous literature [122, 124–126].

Table 5.1: Finite element analysis results of different reconstructed models. Both the top surface area and the bottom fixed support area were restricted around 6.8 $\times 10^{-4}$ m^2. P_1 stands for the equivalent (von-Mises) stress under a follower load of 1000 N and P_2 stands for the von-Mises stress under both the follower load of 1000 N and a bending moment of 7.5 $N \cdot m$. $\Delta P_{1,2}$ shows the absolute error between the stress of the gold standard model and the automatic reconstructed models.

	Gold Standard	**Manual Label**	**SegNet**	**UNet**	**ResUNet**	**KiUNet**
Top Surface	6.803×10^{-4}	6.8014×10^{-4}	6.8086×10^{-4}	6.7977×10^{-4}	6.7986×10^{-4}	6.8058×10^{-4}
Area (m^2)	(2083 Faces)	(1792 Faces)	(1771 Faces)	(1672 Faces)	(1753 Faces)	(1563 Faces)
Bottom Surface	6.803×10^{-4}	6.8043×10^{-4}	6.8004×10^{-4}	6.8066×10^{-4}	6.8047×10^{-4}	6.8082×10^{-4}
Area (m^2)	(1968 Faces)	(1848 Faces)	(1799 Faces)	(1730 Faces)	(1851 Faces)	(1443 Faces)
2D Dice Score			0.9002	0.9333	0.9483	0.9715
3D Dice Score		0.9641	0.882875	0.9243	0.91413	0.927957
3D IOU		0.93078	0.79031	0.859258	0.841841	0.865597
P_1 (Pa)	4.5053×10^{5}	4.1491×10^{5}	4.0931×10^{5}	3.8804×10^{5}	4.6977×10^{5}	4.1305×10^{5}
ΔP_1 (Pa)		3.5620×10^{4}	4.1220×10^{4}	6.2490×10^{4}	1.9240×10^{4}	3.7480×10^{4}
P_2 (Pa)	2.0973×10^{6}	1.9873×10^{6}	2.2020×10^{6}	1.8814×10^{6}	1.9677×10^{6}	1.9450×10^{6}
ΔP_2 (Pa)		1.1000×10^{5}	1.0470×10^{5}	2.1590×10^{5}	1.2960×10^{5}	1.5230×10^{5}

Table 5.1 shows the details of two experiments and the corresponding FEA results. Both the top and bottom surfaces were selected as close as to 6.8 $\times 10^{-4} m^2$ area. $P_{1,2}$ were the results of equivalent (von-Mises) stresses from different models in Experiments 1 and 2. $\Delta P_{1,2}$ computed the absolute error of von-Mises stresses between the gold standard model and other reconstructed models. Fig. 5.2 is the line graph describing the FEA results.

In Experiment 1, considering the results of static model evaluation metrics of 3D Dice score and 3D IOU, the equivalent stress of the model reconstructed from manual label still performed quite close to the gold standard, indicating an accurate reconstruction. The ResUNet vertebra model obtained the closest biomechanical results to the gold standard model - which means although the ResUNet model failed to segment the transverse process, it still had the most accurate top surface area, corresponding to the best validation Dice score during training. The KiUNet model had the second accurate FEA results among the automatic reconstructed models with the second accurate top surface, despite the fact that

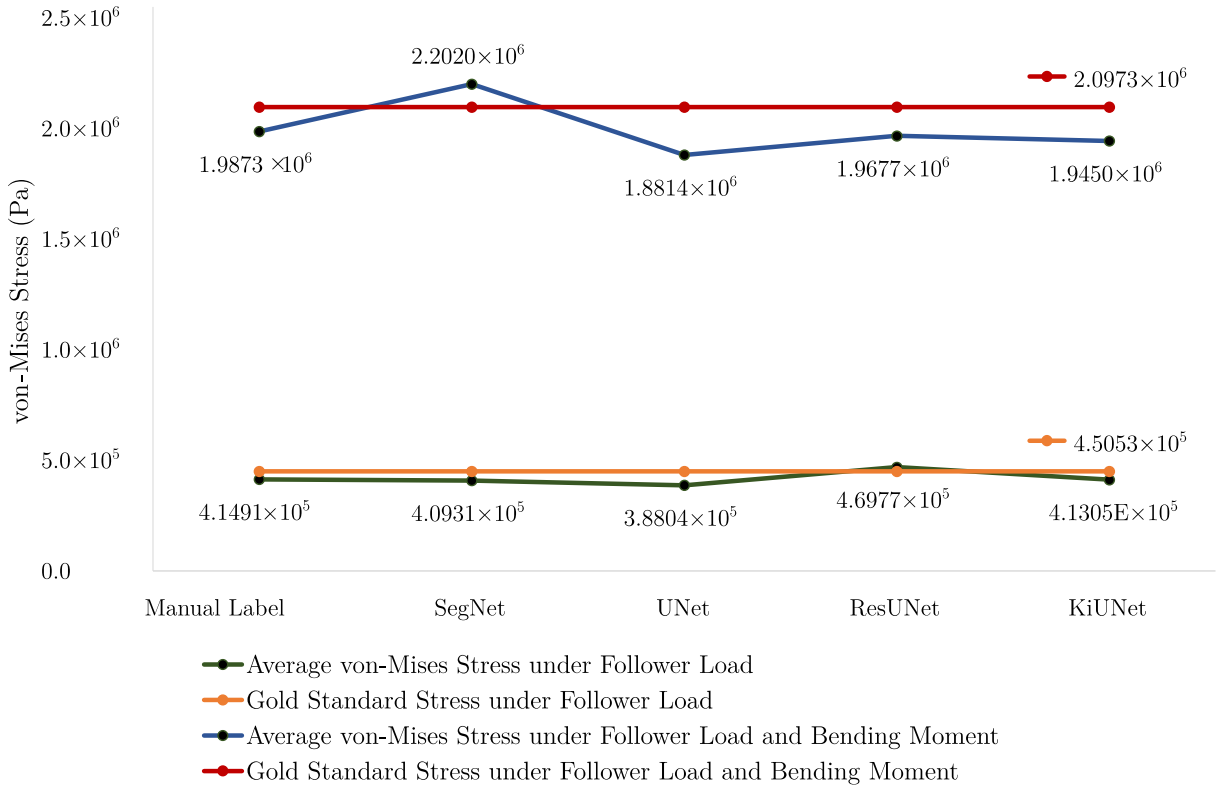


Figure 5.2: The line graph shows the von-Mises stress of different reconstructed models in the scenario of only the follower load and the scenario of both the follower load and the bending moment. Two lines of stress from the FAROArm-scanned model are set as the gold standard.

the KiUNet has the best overall accuracy as it achieved the best 3D Dice score and the 3D IOU score. Following is the SegNet model that has the smallest volume. The UNet model had the worst performance, which mainly contributed to the stripes layer by layer on the surface area, though the UNet model achieved better results than SegNet model in the static 3D evaluation metrics.

In Experiment 2, the FEA results as well as the model form error were further magnified with an extra bending moment, reflecting the top surface quality more clearly. The manual label model still has a very close result to the gold standard model with 1.1 $\times 10^5 pa$ error. The SegNet model has the least absolute error in all the automatic reconstructed models.

However, it is worth noting that the von-Mises stress of the SegNet model is actually larger than the gold standard stress, which is probably because of the smaller volume of the SegNet model; hence, the result may be overrated. The rest results follow the sequence of ResUNet, KiUNet and UNet, which is reasonable compared with the results of 3D evaluation metrics.

The FEA results did not show a strong positive correlation with the static model accuracy. This is mainly because in FEA, the top (or bottom) surface quality is more critical to affecting the biomechanical results than the overall model accuracy. Even if some models reconstructed from neural networks did achieve a better score in the overall accuracy, they may not have a better quality in these partial surface areas.

To conclude, the FEA provides an evaluation method not only presenting the stress distribution but also evaluating the model form error from a different perspective especially for the partial surface accuracy that the static evaluation metrics cannot quantify.

Chapter 6

Discussion and Conclusions

6.1 Discussion

This research proposed a workflow of quick and accurate automatic subject-specific vertebra reconstruction method using four different deep learning models for automatic vertebra segmentation. The workflow performance in a clinical application was validated using an excised human lumbar spine, which was scanned via computed tomography and reconstructed into 3D CAD models using the above refined neural networks. The original geometry of the excised vertebra, also serving as the gold standard model, was obtained using a high-precision reverse engineering solution. Several 3D volumetric evaluation metrics were modified to quantify the overall model accuracy and a finite element analysis method was designed to show the model form errors and to further evaluate the partial surface accuracy from a different perspective.

However, there still exist several limitations during the study:

(1) Due to the GPU memory capacity, the image datasets for training and the CT

scan of the dried-out vertebra were all preprocessed to reduce the resolution. The output segmented masks would then have a lower resolution than the original corresponding CT image, which will affect the final reconstructed model precision.

(2) The CT scanning can have a perspective of the interior vertebra due to the features of X-rays, observing the interior details of the bone structure such as the cancellous bone. However, when the vertebra is reconstructed into 3D surface mesh, the inside information will be lost. The 3D evaluation metrics only focus on the surface accuracy while the FEA will instead mesh the interior structure.

(3) Current FEA methods for evaluation only include the follower load and bending moment applied on a single lumbar vertebra with homogenous material despite the difference between cortical and cancellous bone. In future, a complete spine model of at least two vertebrae and the intervertebral discs with real material properties will be built for simulation closer to the real scenario.

These limitations will be focused on and addressed in future research.

6.2 Conclusion

This research proposed and validated the workflow of a quick and accurate automatic subject-specific vertebra reconstruction method. Four different neural networks, SegNet, UNet, ResUNet and KiUNet were customized and trained for automatic vertebra segmentation with the highest validation Dice score of 0.942 in ResUNet and the highest training Dice score of 0.9505 in KiUNet. The reconstruction errors were quantified in clinical applications via an excised human lumbar spine, which was CT scanned and

reconstructed into 3D CAD models using four refined networks. A reverse engineering solution based on a high-precision measuring robotic arm was proposed to obtain the original geometry of the excised vertebra, serving as the gold standard. Details of 3D evaluation metrics as well as the heatmap/histogram and the results were presented for each reconstructed model. An FEA method was designed to explore potential biomechanical results of different reconstructed models and to further evaluate the partial surface accuracy. Those errors should be considered when leveraging subject-specific modeling towards the development of improvement of treatments.

Bibliography

[1] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *ArXiv e-prints*, mar 2016.

[2] A. L. Williams, A. Al-Busaidi, P. J. Sparrow, J. E. Adams, and R. W. Whitehouse, "Under-reporting of osteoporotic vertebral fractures on computed tomography," *European journal of radiology*, vol. 69, no. 1, pp. 179–183, 2009.

[3] D. Müller, J. S. Bauer, M. Zeile, E. J. Rummeny, and T. M. Link, "Significance of sagittal reformations in routine thoracic and abdominal multislice ct studies for detecting osteoporotic fractures and other spine abnormalities," *European radiology*, vol. 18, no. 8, pp. 1696–1702, 2008.

[4] D. C. Howlett, K. J. Drinkwater, N. Mahmood, J. Illes, J. Griffin, and K. Javaid, "Radiology reporting of osteoporotic vertebral fragility fractures on computed tomography studies: results of a uk national audit," *European Radiology*, vol. 30, no. 9, pp. 4713–4723, 2020.

[5] N. Mitsuhashi, K. Fujieda, T. Tamura, S. Kawamoto, T. Takagi, and K. Okubo, "Bodyparts3d: 3d structure database for anatomical concepts," *Nucleic acids research*,

vol. 37, no. suppl_1, pp. D782–D785, 2009.

[6] S. Wan, B. Xue, and Y. Xiong, "Three-dimensional biomechanical finite element analysis of lumbar disc herniation in middle aged and elderly," *Journal of Healthcare Engineering*, vol. 2022, 2022.

[7] S.-H. Kang, M.-K. Kim, H.-J. Kim, P. Zhengguo, and S.-H. Lee, "Accuracy assessment of image-based surface meshing for volumetric computed tomography images in the craniofacial region," *Journal of Craniofacial Surgery*, vol. 25, no. 6, pp. 2051–2055, 2014.

[8] J. Schmidt, J. Engh, M. Viceconti, and H. Ploeg, "What is the accuracy of surface mod-els created from visible human male computed tomography data," in *Proceedings of the ASB 29th annual meeting*, p. 63, 2005.

[9] F. Gelaude, J. Vander Sloten, and B. Lauwers, "Semi-automated segmentation and visualisation of outer bone cortex from medical images," *Computer methods in biomechanics and biomedical engineering*, vol. 9, no. 1, pp. 65–77, 2006.

[10] M. Loubele, F. Maes, D. Vandermeulen, K. Denis, R. Jacobs, S. White, D. van Steenberghe, A. Van Bael, D. Loeckx, I. Lambrichts, *et al.*, "Assessment of bone segmentation quality of ct scanners using laser scanning," *International Journal of Computer Assisted Radiology and Surgery*, vol. 1, no. SUPPL. 7, pp. 400–402, 2006.

[11] J.-Y. Choi, J.-H. Choi, N.-K. Kim, Y. Kim, J.-K. Lee, M.-K. Kim, J.-H. Lee, and M.-J. Kim, "Analysis of errors in medical rapid prototyping models," *International journal of oral and maxillofacial surgery*, vol. 31, no. 1, pp. 23–32, 2002.

[12] J. Santolaria, R. Jiménez, M. Rada, and F. Loscos, "Error compensation method for improving the accuracy of biomodels obtained from cbct data," *Medical Engineering & Physics*, vol. 36, no. 3, pp. 397–404, 2014.

[13] W. P. Engelbrecht, Z. Fourie, J. Damstra, P. O. Gerrits, and Y. Ren, "The influence of the segmentation process on 3d measurements from cone beam computed tomography-derived surface models," *Clinical oral investigations*, vol. 17, no. 8, pp. 1919–1927, 2013.

[14] M. L. Poleti, T. M. F. Fernandes, O. Pagin, M. R. Moretti, and I. R. F. Rubira-Bullen, "Analysis of linear measurements on 3d surface models using cbct data segmentation obtained by automatic standard pre-set thresholds in two segmentation software programs: an in vitro study," *Clinical oral investigations*, vol. 20, no. 1, pp. 179–185, 2016.

[15] J. Zhang, D. Malcolm, J. Hislop-Jambrich, C. D. L. Thomas, and P. M. Nielsen, "An anatomical region-based statistical shape model of the human femur," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 2, no. 3, pp. 176–185, 2014.

[16] S. Zhou, Y. Cheng, Y. Wang, K. Dong, C. Guo, J. Bai, and S. Tamura, "Segmentation of the hip joint in ct volumes using adaptive thresholding classification and normal direction correction," *Journal of the Chinese Institute of Engineers*, vol. 36, no. 8, pp. 1059–1072, 2013.

[17] N.-F. Tian, Q.-S. Huang, P. Zhou, Y. Zhou, R.-K. Wu, Y. Lou, and H.-Z. Xu, "Pedicle screw insertion accuracy with different assisted methods: a systematic review and meta-analysis of comparative studies," *European Spine Journal*, vol. 20, no. 6, pp. 846–859, 2011.

[18] Y. Abe, S. Sato, K. Kato, T. Hyakumachi, Y. Yanagibashi, M. Ito, and K. Abumi, "A novel 3d guidance system using augmented reality for percutaneous vertebroplasty," *Journal of Neurosurgery: Spine*, vol. 19, no. 4, pp. 492–501, 2013.

[19] J. T. Gibby, S. A. Swenson, S. Cvetko, R. Rao, and R. Javan, "Head-mounted display augmented reality to guide pedicle screw placement utilizing computed tomography," *International journal of computer assisted radiology and surgery*, vol. 14, no. 3, pp. 525–535, 2019.

[20] F. Liebmann, S. Roner, M. von Atzigen, D. Scaramuzza, R. Sutter, J. Snedeker, M. Farshad, and P. Fürnstahl, "Pedicle screw navigation using surface digitization on the microsoft hololens," *International journal of computer assisted radiology and surgery*, vol. 14, no. 7, pp. 1157–1165, 2019.

[21] H. Liu, J. Wu, Y. Tang, H. Li, W. Wang, C. Li, and Y. Zhou, "Percutaneous placement of lumbar pedicle screws via intraoperative ct image–based augmented reality–guided technology," *Journal of Neurosurgery: Spine*, vol. 32, no. 4, pp. 542–547, 2019.

[22] T. M. Urakov, M. Y. Wang, and A. D. Levi, "Workflow caveats in augmented reality–assisted pedicle instrumentation: cadaver lab," *World neurosurgery*, vol. 126, pp. e1449–e1455, 2019.

[23] P. Wei, Q. Yao, Y. Xu, H. Zhang, Y. Gu, and L. Wang, "Percutaneous kyphoplasty assisted with/without mixed reality technology in treatment of ovcf with ivc: a prospective study," *Journal of orthopaedic surgery and research*, vol. 14, no. 1, pp. 1–9, 2019.

[24] F. Wanivenhaus, C. Neuhaus, F. Liebmann, S. Roner, J. M. Spirig, and M. Farshad, "Augmented reality-assisted rod bending in spinal surgery," *The Spine Journal*, vol. 19, no. 10, pp. 1687–1689, 2019.

[25] M. von Atzigen, F. Liebmann, A. Hoch, D. E. Bauer, J. G. Snedeker, M. Farshad, and P. Fürnstahl, "Holoyolo: A proof-of-concept study for marker-less surgical navigation of spinal rod implants with augmented reality and on-device machine learning," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 17, no. 1, pp. 1–10, 2021.

[26] J. Gasco, A. Patel, J. Ortega-Barnett, D. Branch, S. Desai, Y. F. Kuo, C. Luciano, S. Rizzi, P. Kania, M. Matuyauskas, *et al.*, "Virtual reality spine surgery simulation: an empirical study of its usefulness," *Neurological research*, vol. 36, no. 11, pp. 968–973, 2014.

[27] J. Shi, Y. Hou, Y. Lin, H. Chen, and W. Yuan, "Role of visuohaptic surgical training simulator in resident education of orthopedic surgery," *World Neurosurgery*, vol. 111, pp. e98–e104, 2018.

[28] M. B. Gottschalk, S. T. Yoon, D. K. Park, J. M. Rhee, and P. M. Mitchell, "Surgical training using three-dimensional simulation in placement of cervical lateral mass

screws: a blinded randomized control trial," *The Spine Journal*, vol. 15, no. 1, pp. 168–175, 2015.

[29] T. Halic, S. Kockara, C. Bayrak, and R. Rowe, "Mixed reality simulation of rasping procedure in artificial cervical disc replacement (acdr) surgery," in *BMC bioinformatics*, vol. 11, pp. 1–17, Springer, 2010.

[30] J. Cauley, D. Thompson, K. Ensrud, J. Scott, and D. Black, "Risk of mortality following clinical fractures," *Osteoporosis international*, vol. 11, no. 7, pp. 556–561, 2000.

[31] D. P. Anitha, T. Baum, J. S. Kirschke, and K. Subburaj, "Effect of the intervertebral disc on vertebral bone strength prediction: A finite-element study," *The Spine Journal*, vol. 20, no. 4, pp. 665–671, 2020.

[32] A. Roychowdhury, "Application of the finite element method in orthopedic implant design," *Journal of long-term effects of medical implants*, vol. 19, no. 1, 2009.

[33] Y. Lu, Z. Yang, and Y. Wang, "A critical review on the three-dimensional finite element modelling of the compression therapy for chronic venous insufficiency," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 233, no. 11, pp. 1089–1099, 2019.

[34] T. Guan, Y. Zhang, A. Anwar, Y. Zhang, and L. Wang, "Determination of three-dimensional corrective force in adolescent idiopathic scoliosis and biomechanical finite element analysis," *Frontiers in Bioengineering and Biotechnology*, p. 963, 2020.

[35] Y. Zhou, D. Xin, Z. Lei, Y. Zuo, and Y. Zhao, "Comparative three-dimensional finite element analysis of 4 kinds of pedicle screw schemes for treatment of adult degenerative scoliosis," *Medical Science Monitor: International Medical Journal of Experimental and Clinical Research*, vol. 26, pp. e922050–1, 2020.

[36] S. Jia, L. Lin, H. Yang, J. Fan, S. Zhang, and L. Han, "The influence of the rib cage on the static and dynamic stability responses of the scoliotic spine," *Scientific Reports*, vol. 10, no. 1, pp. 1–10, 2020.

[37] R. Basaran, M. Efendioglu, M. Kaksi, T. Celik, İ. Mutlu, and M. Ucar, "Finite element analysis of short-versus long-segment posterior fixation for thoracolumbar burst fracture," *World Neurosurgery*, vol. 128, pp. e1109–e1117, 2019.

[38] R. Zhu, Y. Chen, Q. Yu, S. Liu, J. Wang, Z. Zeng, and L. Cheng, "Effects of contusion load on cervical spinal cord: A finite element study," *Mathematical Biosciences and Engineering*, vol. 17, no. 3, pp. 2272–2283, 2020.

[39] X. Tang, C. Liu, K. Huang, G. Zhu, H. Sun, J. Dai, and J. Tian, "Analysis of a three-dimensional finite element model of atlas and axis complex fracture," *Zhonghua yi xue za zhi*, vol. 98, no. 19, pp. 1484–1488, 2018.

[40] A. A. Gandhi, N. M. Grosland, N. A. Kallemeyn, S. Kode, D. C. Fredericks, and J. D. Smucker, "Biomechanical analysis of the cervical spine following disc degeneration, disc fusion, and disc replacement: a finite element study," *International journal of spine surgery*, vol. 13, no. 6, pp. 491–500, 2019.

[41] X.-Y. Cai, D. Sang, C.-X. Yuchi, W. Cui, C. Zhang, C.-F. Du, and B. Liu, "Using finite element analysis to determine effects of the motion loading method on facet joint forces after cervical disc degeneration," *Computers in biology and medicine*, vol. 116, p. 103519, 2020.

[42] K. Imai, "Computed tomography-based finite element analysis to assess fracture risk and osteoporosis treatment," *World journal of experimental medicine*, vol. 5, no. 3, p. 182, 2015.

[43] H.-Z. Guo, D.-Q. Guo, Y.-C. Tang, D. Liang, and S.-C. Zhang, "Selective cement augmentation of cranial and caudal pedicle screws provides comparable stability to augmentation on all segments in the osteoporotic spine: a finite element analysis," *Annals of translational medicine*, vol. 8, no. 21, 2020.

[44] Q. Fei, Q. Li, Y. Yang, D. Li, H. Tang, J. Li, B. Wang, and Y. Wang, "Three-dimensional finite element model of thoracolumbar spine with osteoporotic vertebral compression fracture," *Zhonghua yi xue za zhi*, vol. 90, no. 41, pp. 2943–2946, 2010.

[45] Y. Lafon, V. Lafage, J.-P. Steib, J. Dubousset, and W. Skalli, "In vivo distribution of spinal intervertebral stiffness based on clinical flexibility tests," *Spine*, vol. 35, no. 2, pp. 186–193, 2010.

[46] A. C. Bourgeois, A. R. Faulkner, A. S. Pasciak, and Y. C. Bradley, "The evolution of image-guided lumbosacral spine surgery," *Annals of Translational Medicine*, vol. 3, no. 5, 2015.

[47] S. Iyer, B. A. Christiansen, B. J. Roberts, M. J. Valentine, R. K. Manoharan, and M. L. Bouxsein, "A biomechanical model for estimating loads on thoracic and lumbar vertebrae," *Clinical biomechanics*, vol. 25, no. 9, pp. 853–858, 2010.

[48] I. El Bojairami, K. El-Monajjed, and M. Driscoll, "Development and validation of a timely and representative finite element human spine model for biomechanical simulations," *Scientific Reports*, vol. 10, no. 1, pp. 1–15, 2020.

[49] E. Newell and M. Driscoll, "The examination of stress shielding in a finite element lumbar spine inclusive of the thoracolumbar fascia," *Medical & Biological Engineering & Computing*, vol. 59, no. 7, pp. 1621–1628, 2021.

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

[51] W. Schroeder, R. Maynard, and B. Geveci, "Flying edges: A high-performance scalable isocontouring algorithm," in *2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 33–40, IEEE, 2015.

[52] Y. Kang, K. Engelke, and W. A. Kalender, "A new accurate and precise 3-d segmentation method for skeletal structures in volumetric ct data," *IEEE transactions on medical imaging*, vol. 22, no. 5, pp. 586–598, 2003.

[53] P. H. Lim, U. Bagci, and L. Bai, "A robust segmentation framework for spine trauma diagnosis," in *Computational Methods and Clinical Applications for Spine Imaging*, pp. 25–33, Springer, 2014.

[54] K. Hammernik, T. Ebner, D. Stern, M. Urschler, and T. Pock, "Vertebrae segmentation in 3d ct images based on a variational framework," in *Recent advances in computational methods and clinical applications for spine imaging*, pp. 227–233, Springer, 2015.

[55] A. Rasoulian, R. Rohling, and P. Abolmaesumi, "Lumbar spine segmentation using a statistical multi-vertebrae anatomical shape+ pose model," *IEEE transactions on medical imaging*, vol. 32, no. 10, pp. 1890–1900, 2013.

[56] D. Štern, B. Likar, F. Pernuš, and T. Vrtovec, "Parametric modelling and segmentation of vertebral bodies in 3d ct and mr spine images," *Physics in Medicine & Biology*, vol. 56, no. 23, p. 7505, 2011.

[57] S. Kadoury, H. Labelle, and N. Paragios, "Automatic inference of articulated spine models in ct images using high-order markov random fields," *Medical image analysis*, vol. 15, no. 4, pp. 426–437, 2011.

[58] B. Ibragimov, R. Korez, B. Likar, F. Pernuš, L. Xing, and T. Vrtovec, "Segmentation of pathological structures by landmark-assisted deformable models," *IEEE transactions on medical imaging*, vol. 36, no. 7, pp. 1457–1469, 2017.

[59] A. Suzani, A. Rasoulian, A. Seitel, S. Fels, R. N. Rohling, and P. Abolmaesumi, "Deep learning for automatic localization, identification, and segmentation of vertebral bodies in volumetric mr images," in *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 9415, p. 941514, International Society for Optics and Photonics, 2015.

[60] C. Chu, D. L. Belavỳ, G. Armbrecht, M. Bansmann, D. Felsenberg, and G. Zheng, "Fully automatic localization and segmentation of 3d vertebral bodies from ct/mr images via a learning-based method," *PloS one*, vol. 10, no. 11, p. e0143327, 2015.

[61] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[62] A. Sekuboyina, A. Valentinitsch, J. S. Kirschke, and B. H. Menze, "A localisation-segmentation approach for multi-label annotation of lumbar vertebrae using deep nets," *arXiv preprint arXiv:1703.04347*, 2017.

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

[64] C. Payer, D. Stern, H. Bischof, and M. Urschler, "Coarse to fine vertebrae localization and segmentation with spatialconfiguration-net and u-net.," in *VISIGRAPP (5: VISAPP)*, pp. 124–133, 2020.

[65] X. Liang, I. Lambrichts, Y. Sun, K. Denis, B. Hassan, L. Li, R. Pauwels, and R. Jacobs, "A comparative evaluation of cone beam computed tomography (cbct) and multi-slice ct (msct). part ii: On 3d model accuracy," *European journal of radiology*, vol. 75, no. 2, pp. 270–274, 2010.

[66] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, pp. 297–302, 1945.

[67] P. Jaccard, "The distribution of the flora in the alpine zone.1," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[68] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[70] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[71] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[72] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[73] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3d deeply supervised network for automatic liver segmentation from ct volumes," in *International conference on medical image computing and computer-assisted intervention*, pp. 149–157, Springer, 2016.

[74] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial*

intelligence and statistics, pp. 249–256, JMLR Workshop and Conference Proceedings, 2010.

[75] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*, pp. 562–570, PMLR, 2015.

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

[77] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 640–651, 2017.

[78] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*, pp. 424–432, Springer, 2016.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.

[80] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, Citeseer, 2013.

[81] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

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

[83] C. Zhang, D. Ai, C. Feng, J. Fan, H. Song, and J. Yang, "Dial/hybrid cascade 3dresunet for liver and tumor segmentation," in *Proceedings of the 2020 4th International Conference on Digital Signal Processing*, pp. 92–96, 2020.

[84] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1451–1460, 2018.

[85] J. M. J. Valanarasu, V. A. Sindagi, I. Hacihaliloglu, and V. M. Patel, "Kiu-net: Overcomplete convolutional architectures for biomedical image and volumetric segmentation," *IEEE Transactions on Medical Imaging*, 2021.

[86] M. Löffler, A. Sekuboyina, A. Jacob, A.-L. Grau, A. Scharr, M. Husseini, M. Kallweit, C. Zimmer, T. Baum, and J. Kirschke, "A vertebral segmentation dataset with fracture grading," *Radiology: Artificial Intelligence*, vol. 2, p. e190138, 07 2020.

[87] J. Yao, J. Burns, D. Forsberg, A. Seitel, A. Rasoulian, P. Abolmaesumi, K. Hammernik, M. Urschler, B. Ibragimov, R. Korez, T. Vrtovec, I. Castro-Mateos, J. Pozo, A. Frangi, R. Summers, and S. Li, "A multi-center milestone study of clinical vertebral ct segmentation," *Computerized Medical Imaging and Graphics*, vol. 49, pp. 16–28, 01 2016.

[88] S.-i. Amari, "Backpropagation and Stochastic Gradient Descent Method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[89] A. Zijdenbos, B. Dawant, R. Margolin, and A. 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.

[90] C. H. Sudre, W. Li, T. K. M. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City, QC,...*, vol. 2017, pp. 240–248, 2017.

[91] K. C. L. Wong, M. Moradi, H. Tang, and T. F. Syeda-Mahmood, "3d segmentation with exponential logarithmic loss for highly unbalanced object sizes," *ArXiv*, vol. abs/1809.00076, 2018.

[92] F. Li, "CS231n: Convolutional neural networks for visual recognition, University of Stanford, Class Lecture." `https://cs231n.github.io/neural-networks-3/#ada`, 2021.

[93] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks* (D. Saad, ed.), Cambridge, UK: Cambridge University Press, 1998. revised, oct 2012.

[94] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[95] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[96] T. Tieleman, G. Hinton, *et al.*, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[97] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[98] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

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

[100] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 249–256, 01 2010.

[101] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[102] F. Girosi, M. Jones, and T. Poggio, "Regularization Theory and Neural Networks Architectures," *Neural Computation*, vol. 7, pp. 219–269, 03 1995.

[103] L. Prechelt, *Early Stopping — But When?*, pp. 53–67. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[104] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint*, vol. arXiv, 07 2012.

[105] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[106] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[107] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[108] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[109] G. van Rossum, "Python tutorial," Tech. Rep. CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.

[110] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[111] FaroTechnologies, "Technical specification sheet for the edge faroarm and scanarm." `https://knowledge.faro.com/Hardware/FaroArm_and_ScanArm/FaroArm_and_ScanArm/Technical_Specification_Sheet_for_the_Edge_FaroArm_and_ScanArm`, 2021.

[112] A. Carass, S. Roy, A. Gherman, J. C. Reinhold, A. Jesson, T. Arbel, O. Maier, H. Handels, M. Ghafoorian, B. Platel, A. Birenbaum, H. Greenspan, D. L. Pham, C. M. Crainiceanu, P. A. Calabresi, J. L. Prince, W. R. G. Roncal, R. T. Shinohara, and I. Oguz, "Evaluating White Matter Lesion Segmentations with Refined Sørensen-Dice Analysis," *Scientific Reports*, vol. 10, p. 8242, May 2020.

[113] M. Polak, H. Zhang, and M. Pi, "An evaluation metric for image segmentation of multiple objects," *Image and Vision Computing*, vol. 27, no. 8, pp. 1223–1227, 2009.

[114] D.-G. Sim, O. Kwon, and R.-H. Park, "Object matching algorithms using robust hausdorff distance measures," *Image Processing, IEEE Transactions on*, vol. 8, pp. 425 – 429, 03 1999.

[115] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[116] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: measuring errors between surfaces using the hausdorff distance," in *Proceedings. IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 705–708 vol.1, 2002.

[117] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992. Range Image Understanding.

[118] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[119] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit*. Kitware, 2006.

[120] S. Naoum, A. V. Vasiliadis, C. Koutserimpas, N. Mylonakis, M. Kotsapas, and K. Katakalos, "Finite element method for the evaluation of the human spine: A literature overview," *Journal of Functional Biomaterials*, vol. 12, no. 3, p. 43, 2021.

[121] A. G. Patwardhan, R. M. Havey, K. P. Meade, B. Lee, and B. Dunlap, "A follower load increases the load-carrying capacity of the lumbar spine in compression," *Spine*, vol. 24, no. 10, pp. 1003–1009, 1999.

[122] A. Rohlmann, L. Bauer, T. Zander, G. Bergmann, and H.-J. Wilke, "Determination of trunk muscle forces for flexion and extension by using a validated finite element model of the lumbar spine and measured in vivo data," *Journal of biomechanics*, vol. 39, no. 6, pp. 981–989, 2006.

[123] A. Rohlmann, T. Zander, M. Rao, and G. Bergmann, "Realistic loading conditions for upper body bending," *Journal of Biomechanics*, vol. 42, no. 7, pp. 884–890, 2009.

[124] V. K. Goel, B. Monroe, L. Gilbertson, and P. Brinckmann, "Interlaminar shear stresses and laminae separation in a disc: finite element analysis of the l3-l4 motion segment subjected to axial compressive loads," *Spine*, vol. 20, no. 6, pp. 689–698, 1995.

[125] V. K. Goel, S. A. Ramirez, W. Kong, and L. G. Gilbertson, "Cancellous Bone Young's Modulus Variation Within the Vertebral Body of a Ligamentous Lumbar Spine—Application of Bone Adaptive Remodeling Concepts," *Journal of Biomechanical Engineering*, vol. 117, pp. 266–271, 08 1995.

[126] A. Shirazi-Adl, A. M. Ahmed, and S. C. Shrivastava, "Mechanical response of a lumbar motion segment in axial torque alone and combined with compression.," *Spine*, vol. 11, no. 9, pp. 914–927, 1986.