Vítor Renato Pires Ferreira

AUTOMATIC LANDMARK DETECTION IN 3D REPRESENTATION OF ORTHOPEDIC IMPLANTS

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Prof. António Dourado Pereira Correia, developed at PeekMed® with the guidance of Eng. Ana Costa, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2023

This page is intentionally left blank.

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Vítor Renato Pires Ferreira

Automatic Landmark Detection In 3D Representation Of Orthopedic Implants

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Prof. António Dourado Pereira Correia, developed at PeekMed® with the guidance of Eng. Ana Costa, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2023

This page is intentionally left blank.

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Vítor Renato Pires Ferreira

Deteção Automática De Pontos De Referência Em Representações 3D De Implantes Ortopédicos

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Sistemas Inteligentes, orientada pelo Professor Doutor António Dourado Pereira Correia, desenvolvida na PeekMed®, com a orientação da Eng. Ana Costa, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho 2023

This page is intentionally left blank.

Acknowledgements

Quero agradecer ao Professor Doutor António Dourado Pereira Correia pela orientação durante o desenvolvimento desta tese, o suporte e conselhos que foram essenciais para a conclusão da mesma. Além disso, quero agradecer à Universidade de Coimbra pelo conhecimento que me transmitiu e que será essencial para a minha vida profissional, assim como a todos os amigos e colegas que me ajudaram ao longo do meu percurso académico.

Quero também agradecer à Eng. Ana Costa e ao Eng. Pedro Dias, pela valiosa ajuda e suporte, assim como a todos os meus colegas na PeekMed®, se mostraram disponíveis para ajudar no que quer que fosse, permitindo conciliar com equilíbrio a minha vida de profissional, enquanto trabalhador da PeekMed®, com os meus estudos, concluindo com sucesso a minha tese.

A todos os meus amigos que sempre me apoiaram, que me incentivaram a continuar e sentiram a minha ausência quando faltava aos cafés e jantares nos dias em que ficava em casa a estudar. Estiveram comigo durante todo o meu percurso académico, partilhei com eles todas as minhas conquistas e derrotas, aturaram-me todos os fins-de-semana e ainda assim continuam comigo. Um obrigado muito especial a cada um deles.

Por último, um obrigado do fundo do coração a toda a minha família e à minha namorada, que sempre estiveram do meu lado. Em todos os momentos em que pensei desistir, tinham sempre uma palavra de força para me dar, acreditaram em mim, quando eu mesmo não acreditava. Foi esse carinho, dedicação, educação, a fé que depositaram em mim, esse amor, que me fizeram continuar e todas as minhas conquistas devem-se a isso. Os valores que me transmitiram ao longo da vida fazem de mim o homem que sou hoje e uma melhor versão de mim que quero ser amanhã. Obrigado por estarem presentes na minha vida, sinto-me grato por partilhar esta etapa da minha vida convosco, assim como as que virão.

This page is intentionally left blank.

Abstract

The planning of a surgery is essential for its success. One of the phases is the detection of anatomical landmarks in order to aid planning. With advances in software and medical imaging technologies, applications have emerged that can increase the accuracy of surgical planning.

The process of manually detecting anatomical landmarks is time consuming and very dependent on the experience of the physician. With advances in Deep Learning (DL) techniques, namely Convolutional Neural Networks (CNN), this process can be automated.

Total Hip Arthroplasty (THA) is known to provide reliable results, improving the patient's quality of life. Implant placement is one of the steps in this procedure, and implant landmarks are needed to position the implant correctly. Usually, this process of manually annotating the implant landmarks is performed by specialists, involving detailed analysis of the implant geometry and marking the relevant points and axes. This is a time-consuming process, and a mechanism to automatically detect these landmarks is helpful.

Just as CNNs can be used to detect anatomical landmarks, the process of detecting landmarks in orthopedic implants can be automated using CNNs. This thesis presents three approaches that use CNNs to automatically detect landmarks in stems, an orthopedic implant belonging to the hip implant system. Two of the three approaches, a CNN inspired by a scientific paper used to detect objects in dilated perivascular spaces, and a ResNet-34, use volumetric data as input data. The third approach, PointNet, uses point clouds as input data. Of all the approaches, PointNet was the best performing, with an Root Mean Square Error (RMSE) of $0.857mm \pm 0.452mm$, a relatively low error considering the average length of a stem.

Keywords

Deep Learning; Landmarks; Total Hip Arthroplasty; Implants; Convolutional Neural Networks.

This page is intentionally left blank.

Resumo

O planeamento de uma cirurgia é essencial para o seu sucesso. Uma das fases é a deteção de pontos de referência anatómicos para auxiliar o planeamento. Com os avanços no software e nas tecnologias de imagem médica, surgiram aplicações que podem aumentar a precisão do planeamento cirúrgico.

O processo de deteção manual de pontos de referência anatómicos consome muito tempo e é muito dependente da experiência do médico. Com os avanços nas técnicas de *Deep Learning*, nomeadamente as Redes Neuronais Convolucionais (RNC), este processo pode ser automatizado.

A Artroplastia Total da Anca (ATA) é conhecida por proporcionar resultados fiáveis, melhorando a qualidade de vida do doente. A colocação do implante é uma das etapas deste procedimento, e são necessários pontos de referência para posicionar corretamente o implante. Normalmente, este processo de anotação manual dos pontos de referência do implante é efectuado por especialistas, envolvendo uma análise detalhada da geometria do implante e a marcação dos pontos e eixos relevantes. Trata-se de um processo que consome bastante tempo, pelo que é útil um mecanismo para detetar automaticamente estes pontos de referência.

Tal como as RNCs podem ser utilizadas para detetar pontos de referência anatómicos, o processo de deteção de pontos de referência em implantes ortopédicos pode ser automatizado utilizando RNCs. Esta tese apresenta três abordagens que utilizam RNCs para detetar automaticamente pontos de referência em hastes, um implante ortopédico pertencente ao sistema de implantes da anca. Duas das três abordagens, uma RNC inspirada num artigo científico utilizado para detetar objectos em espaços perivasculares dilatados, e uma ResNet-34, utilizam dados volumétricos como dados de entrada. A terceira abordagem, PointNet, utiliza nuvens de pontos como dados de entrada. De todas as abordagens, a PointNet obteve o melhor desempenho, com um RMSE de $0.857mm \pm 0.452mm$, um erro relativamente baixo tendo em conta o comprimento médio de uma haste.

Palavras-Chave

Deep Learning; Pontos de referência; Artroplastia Total da Anca; Implantes; Redes Neuronais Convolucionais.

This page is intentionally left blank.

Contents

Acronyms

ANN Artificial Neural Network.

ATA Artroplastia Total da Anca.

CBCT Cone Beam Computed Tomography.

CNN Convolutional Neural Networks.

CPM Convolutional Pose Machines.

CSV Comma-separated values.

CT Computed Tomography.

DL Deep Learning.

LPN Lightweight Pose Network.

ML Machine Learning.

MLP Multilayer Perceptron.

MRI Magnetic Resonance Imaging.

MSE Mean Square Error.

RGB Red Green Blue.

RMSE Root Mean Square Error.

RNC Redes Neuronais Convolucionais.

SOTA State Of The Art.

THA Total Hip Arthroplasty.

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

This chapter gives a short introduction to surgical planning, the importance of anatomical landmarks detection in this process and the automatic detection mechanisms of anatomical landmarks. This chapter also describes orthopedic procedures such as Total Hip Arthroplasty (THA) and implant systems, introduces PeekMed®, the company where the thesis was developed, as well as describing the problem and objectives of this thesis.

1.1 Overview

The preoperative planning process is crucial for the success of a surgical intervention. With the advances in software and medical imaging, planning software applications have surfaced and become popularized as a way to increase accuracy and efficiency to the surgical planning process. One of the most common steps in surgical planning is the anatomical landmarking. An anatomical landmark is a point or structure, used as a reference point to identify and describe regions or structures in the human body [1].

In the field of orthopedic surgery, anatomical landmarking is a common practice that allows morphological parameters to be calculated. With these measurements, it is possible to identify possible malformed morphologies in the patient and use them to help simulate an orthopedic procedure or the placement of an orthopedic implant. When it comes to identifying anatomical landmarks, most software applications, like Mimics®, that work with medical imaging technologies, such as X-ray, Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), require the surgeon to manually identify landmarks, but it remains an exhaustive process that is heavily influenced by the surgeon's observation, knowledge, and experience [2].

THA is an orthopedic procedure that consists of separating the femoral head and neck and removing the acetabulum, replacing the hip joint with an implant system. The hip is a joint made up of the femoral head and the acetabulum (as shown in Figure 1.4). A femoral head is the upper part of the femur. The acetabulum is a cavity in the pelvis where the femoral head fits. This joint is protected by a set of resistant muscles and allows movement in all axes, supporting the human body mass and providing stability to the body during movement [3].

A hip implant system consists of up to four components: cup, stem, liner, and head (Figure 1.2). This set of components works as follows: the stem fits into the femur and is connected to the head, the head together with the liner are placed inside the cup, replacing the acetabulum (Figure 1.3) [4].

THA is a cost-effective and consistently successful surgery for patients suffering from end-stage degenerative osteoarthritis of the hip, a disease that affects millions of people, mainly the elderly, and is the main underlying diagnosis leading to THA. This procedure provides reliable outcomes such as pain relief, functional recovery and improved quality of life [5]. The hip replacement market is expanding globally, with millions of surgeries performed each year and projected market growth in the coming years [6–8].

Anatomical landmarks play an important role in one of the main planning steps of orthopedic surgery, like THA, as well as the placement of an implant. Software applications often aim to streamline this step by automating the placement. For the automated implant placement process, the patient's anatomical landmarks are needed, as well as landmarks and axial definition of the implant itself, in order to be able to fit the implant into the bone [9]. The process of annotating landmarks on implants is usually performed manually by specialists and consists of analyzing the implant geometry and annotating the relevant landmarks and axes. The process of manually annotating landmarks, whether in implants or in anatomical regions, is a considerable time-consuming and expensive process [10], which is a problem for surgical planning software companies that need to perform landmark annotations for the purpose of generating accurate pre-operative plans.

The process of anatomical landmark detection can be automated by software using Deep Learning (DL) techniques and medical images as input. The use of a CNN, a network architecture for DL, allows the software to learn how to recognize relevant patterns and features in the images, enabling it to accurately identify the desired anatomical landmarks. These automated approaches offer an efficient and accurate alternative, as well as for medical treatment planning [11].

Just as approaches based on CNN are used for the automatic detection of anatomical landmarks, it could also be applied to the detection of landmarks on orthopedic implants. By using 3D objects representing the implants, as well as the corresponding landmarks for each implant as input for the CNN models, it would be possible to train the models to accurately identify the relevant landmarks on the implants. This could simplify and speed up the surgical planning process, offering an automated and reliable approach to the correct placement of orthopedic implants.

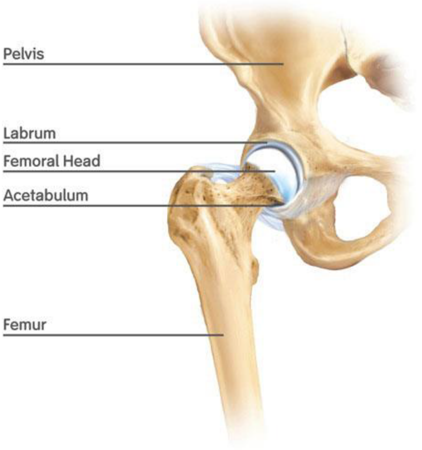


Figure 1.1: Hip anatomy, composed by the pelvis, labrum, femoral head, acetabulum and femur [12].

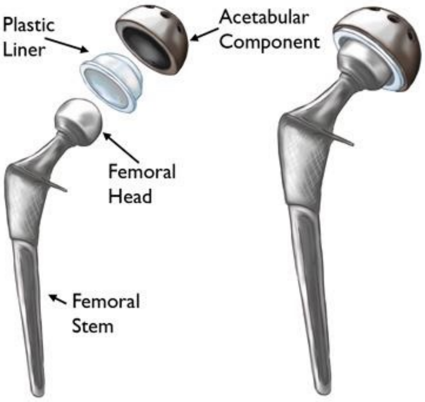


Figure 1.2: Hip implant system components [13].

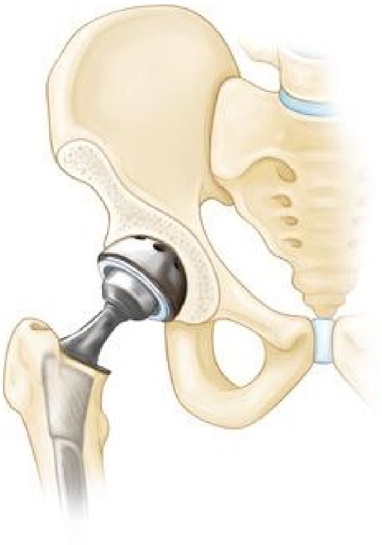


Figure 1.3: Fitting between the hip implant system and the hip joint [13].

1.2 PeekMed®

Figure 1.4: Company logo.

PeekMed® was established in 2015 when three biomedical engineers started the development of a product with the same name based on an idea that originated from the master's thesis of the company's co-founder and CEO (masters in Biomedical Engineering, University of Minho). The goal was to provide orthopedic surgeons with a tool that assists them with a rigorous and objective pre-operative planning software that was capable of making surgeries more predictable, effective and safer for patients.

From the idea a prototype was developed to optimize the whole surgical workflow and to understand which tools were the most important for the user, based on 30+ orthopedic surgeons feedback. And in the beginning of 2015, PeekMed® v1.0 was launched: PeekMed® was able to develop an established pre-operative planning system that obtained the medical device certifications to be commercialized in Europe, Brazil and the United States, where it gained strong commercial traction.

Since then, PeekMed® has adopted the overall mission to develop innovative technical solutions for pre-operative planning of orthopedic surgeries, that are able to fully understand the patient's problem, selecting surgical materials and dimensions, anatomical corrections required, and basically everything about the intervention in seconds.

The current version of PeekMed® has been developed to be a 3D AI-based pre-operative planning system for orthopedic surgery that allows surgeons to plan the procedure easily and automatically using the patient's X-ray or CT, without the need of an engineer. By using an AI-based system, PeekMed® speeds and automatizes several time-consuming and cumbersome steps by performing automatic bone segmentation, automatic anatomical landmark detection, and the automatic planning of the procedure, using the best surgical practices and quick selection of the correction and implants needed.

PeekMed® can automatically perform the pre-operative planning in a matter of seconds, including the accurate prediction of the size and type of implant and tools to be used in surgery, which minimizes the quantity of implants needed to be consigned in each hospital and the number of tools sterilized for the procedure.

1.3 Objectives

Implant placement is one of the features of PeekMed® software. To start surgical planning, the user (orthopedic surgeon) opens a medical image in the PeekMed® software, where it is possible to take the essential measurements for the surgical procedure. Then the procedure is performed, where anatomical landmarks are marked and measurements are generated to aid planning. Next, there may be the need to place a system of implants, and it is necessary to know where each implant is placed and how they connect to each other (using the positioning of certain anatomical landmarks and lines defined by the surgical procedure performed previously).

In THA (a procedure that can be performed with PeekMed® software), up to four implants (cup, stem, liner and head) have to be fitted. After the anatomical landmarks are marked, the implants are placed, starting with the cup that is automatically placed according to certain anatomical landmarks previously marked. After the cup is placed, the stem is automatically placed using the landmarks of the cup as reference (the same applies to the liner and the head).

The benefits of preoperative planning of THA are several and can include the considerable improvement of surgical outcomes and decrease the time duration of surgery, in addition to facilitating precise, efficient, and safe THA [14]. Preoperative planning allows the prediction of several aspects of surgery, including anticipation of equipment needs, identification of surgical access points, as well as prior knowledge of a patient's specific anatomy [15; 16]. Moreover, successful preoperative templating allows the surgeon to assess the size and orientation of implants to be used during operation, which prevents the use of undersized or oversized hip implants intraoperatively. The capability to select the correct size of implants not only considerably reduces the number of intraoperative errors related to template sizing but is also determinative of the long-term success of surgery [17].

Results from inadequate surgical planning and from preoperative templating may include limb length inequality and/or instability, fractures, and other complications that can have damaging and serious effects to patients [14; 18]. Surgical planning in THA is pivotal for the determination of correct implant size, alignment and fit, for correct restoration of the hip.

Given that DL techniques can be used to automatically detect anatomical landmarks [11] and the importance of orthopedic implant landmarks in surgery planning, this study aims to investigate approaches for automatic landmark detection in orthopedic implants. DL techniques will be explored, with a focus on CNN, using architectures described in relevant scientific papers as a basis.

To perform this research, a dataset consisting of a set of 3D models of orthopedic implants will be used, along with the respective landmarks of each implant, which will be analyzed and pre-processed before being used by the neural networks. Subsequently, a comparative analysis of the different approaches will be performed to identify which architecture is most effective in this task.

In summary, the objectives can be outlined as follows:

- data analysis to check which implants are valid and which are invalid;
- adapting the data to use as input for the different approaches;
- checking whether corrections need to be applied to the input data;
- investigate CNN architectures in scientific papers;
- check whether it is necessary to adapt CNN architectures to the data;
- train neural networks;
- analysis of results.

1.4 Document structure

This section provides an overview of the document's content, outlining the chapters and subjects covered in each respective section.

- **Chapter One** - Corresponds to the current chapter. It contains an introduction to contextualize the problem, the motivations for doing the thesis, the company where the thesis will be carried out, and the objectives of the thesis.
- **Chapter Two** - This chapter contains a State Of The Art (SOTA) in techniques applied to problems similar to the problem described in this thesis. It presents several methods of orthopedic surgical planning, the importance of anatomical landmarks in this process and mechanisms for the automatic detection of these landmarks, namely Machine Learning (ML) and DL techniques, which will serve as inspiration for the automatic detection of landmarks in orthopedic implants.
- **Chapter Three** - Chapter where the architectures of DL neural networks will be presented, as well as a description of the data and how the analysis and pre-processing of these data was performed.
- **Chapter Four** - Chapter where the results of training the neural networks will be presented and discussed.
- **Chapter Five** - This chapter presents the conclusion and suggestions for future work.

Chapter 2

State Of The Art

This chapter presents mechanisms for automatic detection of anatomical landmarks, which can serve as inspiration for the development of similar methods applied to automatic landmark detection in orthopedic implants. The various methods for automatic detection of anatomical landmarks are described, with a focus on Deep Learning (DL) techniques, with a introduction to Convolutional Neural Networks (CNN) and relevant examples of studies in which CNNs have been employed in an attempt to perform automatic detection of anatomical landmarks.

Manually marking of anatomical landmarks is a time-consuming process that relies heavily on the experience and knowledge of the healthcare professional. Even today there are situations where manual marking is preferred over automatic marking, due to precision and reliability, as in the case of radiographs, 2D images, of the craniofacial anatomy [19]. When moving from 2D to 3D, the complexity naturally increases, which leads to an even longer process compared to marking landmarks on 2D images, an automatic anatomical landmark detection system would be the perfect solution to speed up the process. The automatic detection of anatomical landmarks is a relevant problem in the scientific community. In recent years, researchers have been developing reliable and accurate methods to detect anatomical landmarks [20; 21].

The automatic anatomical landmark detection methods can be divided into: knowledge based landmarking, atlas based landmarking and learning based landmarking [22].

2.1 Knowledge-based landmarking

An example of landmark detection method using knowledge based was published in the scientific article "Knowledge-Based Landmarking of Cephalograms" [23]. In this study, the followed strategy began with pre-processing the image (in the study in question, cephalograms). The lines and edges contained in the medical images were highlighted, using pre-filtering, resorting to a median filter to remove noise and later an edge detector, using the Mero-Vassy operator, a simplified version of the Hueckel edge detector. The Hueckel edge detector works by approximating the input image by a series of orthogonal functions, where the precision and complexity of the process are determined by the number of terms in the expansion [24].

After pre-processing, line extraction was performed. A knowledge-based line-follower algorithm was applied, which was divided into three categories: the approximate location of the lines, the conditions under which it starts and ends, and the number of segments that constitute it. A priori knowledge implemented in the algorithm must consider the variation of lines and shapes, that vary from patient to patient. Finally, the landmarks are marked, which after extracting the lines, is a direct process.

A knowledge-based method does not require a lot of computational power and conforms to agreed mathematical and anatomical definitions. However, this method is heavily dependent on image quality and therefore is sensitive to noise, has difficulties in identifying landmarks on curved surfaces, contour detection tends to generate errors and if a patient has a significant anomaly (lines and shapes differ of the standard human anatomy), makes the detection of landmarks even more difficult.

2.2 Atlas-based landmarking

In "Atlas-Based Recognition of Anatomical Structures and Landmarks to Support the Virtual Three-Dimensional Planning of Hip Operations" [25], an atlas-based solution for automatic anatomical landmark detection is presented.

The first step is to transfer information from the atlas to the patient image dataset, using a totally freeform registration process. Next, a rough automatic segmentation of the patient's bones is done using threshold-based methods and morphological operators. The label of the nearest structure in the transformed atlas dataset is assigned to each segmented voxel of the patient's bones using a nearest-neighbor approach. This process allows for bony patient voxels not covered by an atlas label to be added to the nearest bone structure.

Images are composed of various levels of gray, and due to various anatomical aspects, such as malformations or bone wear, these gray levels vary. These variations lead to an inaccuracy in transferring landmarks from an atlas to patient data. Therefore, a surface-based recording method was developed that ensures greater robustness and accuracy in transferring this data.

In its initial phase, the Marching-Cubes algorithm is used to generate surface models of the patient's bone structures. Marching-Cubes is an algorithm that creates triangular models of surfaces of constant density from 3D medical data, composed of multiple 2D slices, and generates a connection between these slices using a divide and conquer approach [26]. Later, to calculate the position of the landmark in the patient data, a non-linear match is made between the atlas surface model and the patient surface model in a local landmark environment.

Atlas-based methods have the advantages of being easy to customize and not requiring a large database. However, in order to obtain more accurate results, it is suggested to have a great variability and precision of the patients' data, as well as a correct choice of the point recording technique, which also has a great impact on the performance of this method.

2.3 Learning-based landmarking

Learning based methods use datasets to learn and, in this case, automatically detect landmarks. These methods can be divided into two categories: statistical or based on Machine Learning (ML)

Compared to knowledge-based and atlas-based, learning-based methods are less sensitive to noise and has the advantage of being able to incorporate the expertise and knowledge of experts using training datasets and locating landmarks on 2D images. However, there are still challenges in applying these methods to 3D imaging, mainly due to the large number of datasets required, which is an issue due to legal and ethical restrictions on medical data. Also, the learning process can be difficult due to the curse of dimensionality when processing high-dimensional images. These challenges represent a significant obstacle to the development of highly accurate and automatic 3D landmark systems [27].

2.3.1 Statistical based

Object detection is a fundamental problem in the area of computer vision and statistical detection learning methods have shown effectiveness and efficiency [28].

Statistical object detection learning methods consist of visual learning of annotated models. Because it is a learning-based method, a dataset is needed to train the model, consisting of sample and background images. Considering a general framework of statistical learning-based object detection, in the training phase, the input images go through a feature representation phase, where the raw pixels of the image are mapped into discriminant feature space, connecting low-level pixels with input from learning/inference algorithms, and later statistical learning is applied, generating trained object models. In the detection phase, the input images are mapped in the same feature representation space and with the introduction of object models trained in the previous phase, it is possible to check if there are objects in the input images and obtain their location [29].

2.3.2 ML and DL based

ML is a subfield of artificial intelligence that employs statistical techniques to create predictions and it is based on the idea that systems can learn from data, identify patterns and make decisions. ML is increasingly common in the health area [30], particularly when it comes to medical imaging, as it is useful for automatically detecting fractures or injuries in the human body, which sometimes have shapes that make it difficult for even a doctor to understand what type of object in the image.

DL is a type of ML inspired by the human brain to learn using data. One main advantage is that, in DL, there is no two-step pipeline with feature engineering and model training, there is only one step that is a mixture of the two, where the model itself learns to do the necessary feature engineering. The advantage is to avoid domain errors, that is, as there is no need to create features, the error that comes from this process is removed.

CNN is a type of Artificial Neural Network (ANN), mainly used for pattern recognition in images. ANNs are computational systems that are inspired by the way biological nervous systems operate. They consist of a large number of interconnected computational nodes (neurons) that work together to learn from input and optimize output. The basic structure of an ANN is made up of an input layer, hidden layers, and an output layer. The main limitation of traditional forms of ANNs is that they struggle with the computational complexity required to compute image data [31].

The CNN architecture consists of a stack of layers, generally divided into three groups of layers: convolutional, pooling, and fully connected. The convolutional layer consists of a collection of convolutional filters that are used to generate the output feature map from the input image. The pooling layer is used for subsampling the feature maps and reducing their size while maintaining the dominant information. Activation function is used to map the input to the output. Fully connected layer is located at the end of the CNN architecture and it is used as a classifier [32].

2.4 DL on anatomical landmark detection

Medical imaging modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) or X-rays are widely used by health professionals to analyze the patient's health status. Technologies such as these have supported advances over the decades, as has state-of-the-art computational analysis of medical images, with DL playing a crucial role in these advances, using techniques such as CNN [33].

Historical methods for automated image classification were time-consuming and required extensive rule-based algorithms or manual feature creation, but with the rise and success of CNNs, decreasing computational costs and powerful graphics processing units available, it is now possible to analyze 3D medical images using 3D DL [34].

The next sections present studies where mechanisms for automatic detection of anatomical landmarks have been developed using CNNs.

2.4.1 2D CNN - Triplane architecture

In "Automated Anatomical Landmark Detection on Distal Femur Surface Using Convolutional Neural Network" [21] a triplane architecture of CNNs is presented. This architecture is used for segmentation of the femoral surface and detection of anatomical landmarks in the femur using 3D medical images. The experimental results provide evidence that the proposed method is successful, time-efficient, and reliable in segmenting the femur and accurately identifying anatomical landmarks.

In the training phase, all 3D images were converted into three sets of 2D images, each set corresponding to the X, Y, Z axes, respectively. For each anatomical landmark, the 2D images are annotated as positive or negative, depending on the presence of the landmark in the image. Thus, 3*n* labels are created (*n* corresponds to the number of landmarks to be detected). To train the 3*n* CNN classifiers, both images and corresponding labels are necessary.

In the test phase, the conversion of 3D images to 2D images is performed again and for each axis of each landmark, the corresponding CNN classifier is applied to obtain a probability distribution of 2D images. The final position of each landmark is the result of combining the three classifiers. Figure 2.1 presents the architecture implemented in this scientific article.

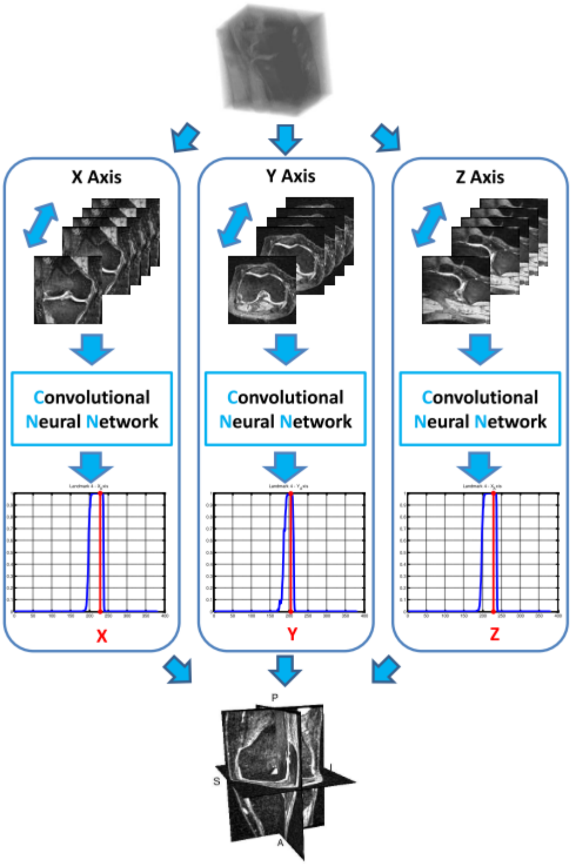

X Axis
Y Axis
Z Axis
Convolutional Neural Network
Convolutional Neural Network
Convolutional Neural Network
X
Y
Z
P
S
A

Figure 2.1: Triplane 2D CNN architecture [21].

2.4.2 2D CNN - Lightweight Pose Network

The thesis "Automatic Anatomical Landmark Location Estimation in Orthopedics" [35] presents a method for automatic detection and location of anatomical landmarks from Red Green Blue (RGB) images of the knee, acquired during a surgery. The results obtained show promising performance, evidencing the ability to generalize to data not seen during model training and provide reliable predictions.

To estimate the location of anatomical points, a Lightweight Pose Network (LPN) [36] architecture, adapted to the described problem, was implemented. The LPN follows an encoder-decoder methodology. The encoder is composed of a ResNet backbone as the main body with a lightweight bottleneck block in the downsampling process, to decrease the number of network parameters. For upsampling, it uses two group deconvolutional layers and a 1x1 convolutional layer, forming the decoder. Figure 2.2 demonstrates the implemented architecture.

The LPN receives as input images cropped using a bounding box of the exposed bone and with a fixed size (250x250). The LPN encodes the input into low-resolution feature representations and uses a decoder to estimate the heatmaps. In the context of the described problem, each heatmap corresponds to the prediction of a different anatomical keypoint.

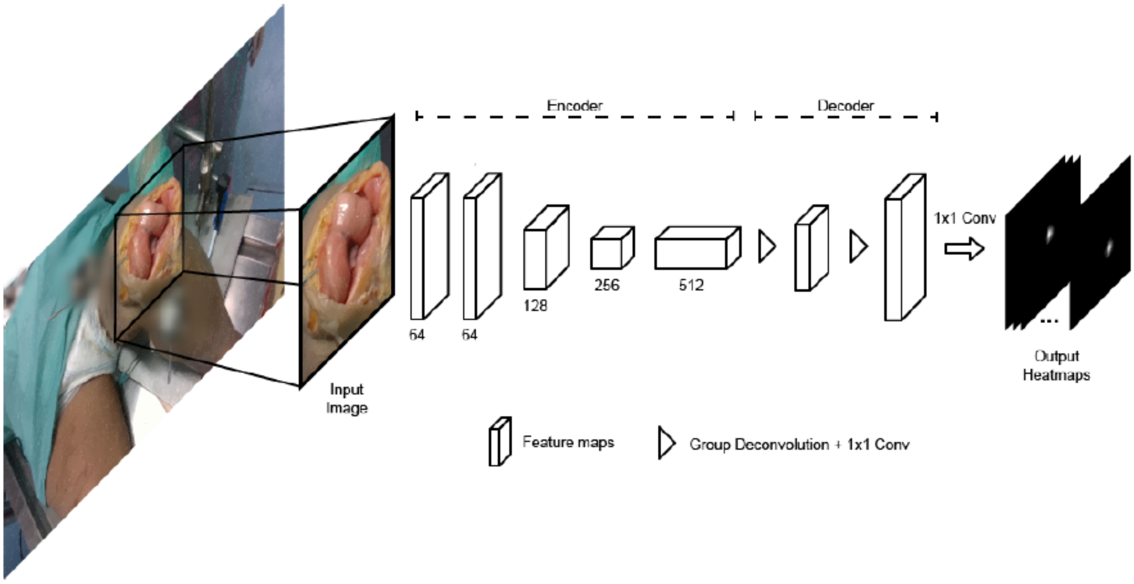


Figure 2.2: LPN architecture [35].

2.4.3 3D CNN

In "Fast and Accurate Craniomaxillofacial Landmark Detection via 3D Faster R-CNN" [37] an architecture of a 3D CNN capable of detecting craniomaxillofacial landmarks is presented. This architecture interprets this task as a generic object detection problem, using Cone Beam Computed Tomography (CBCT) images (which is volumetric data resulting from a cone beam of X-rays to obtain a series of projections of various positions around the patient) and landmarks manually digitized by experienced craniomaxillofacial surgeons. The study achieves state-of-the-art performance in efficiently localizing a varying number of craniomaxillofacial landmarks on a CBCT dataset.

3D faster R-CNN (Figure 2.3) is an architecture resulting from the combination of three CNN architectures: a backbone, a 3D RPN and a 3D fast R-CNN.

The backbone is a CNN that has the function to extract features from the input 3D volumes, allowing the model to capture relevant spatial information. The backbone has lateral connections in its architecture that allow information to flow between layers of different scales, so that higher layers can access features from lower levels, forming a feature pyramid that has feature maps (representations of the features extracted at different levels of the pyramid).

The 3D RPN, a small network shared by the feature maps in the feature pyramid, uses anchors, predefined bounding boxes that are placed at various locations in the 3D volume, to generate the proposed regions of interest that are likely to contain landmarks. For each anchor, the 3D RPN evaluates the likelihood of containing a landmark within it using a combination of classification, which determines whether the anchor contains a landmark, and regression, which adjusts the spatial coordinates of the anchor to best align with the exact location of the landmark.

The 3D fast R-CNN receives as input the regions of interest generated by the 3D RPN to determine the position of the landmarks. Initially a region pooling is performed, where the extracted regions of interest are mapped to a fixed representation using a pooling operation, transforming each region of interest into a feature vector with a fixed size. Next is classification, where from the feature vectors of the regions of interest, Fast R-CNN uses a fully connected neural network to perform the classification of the landmarks present in the regions of interest. Finally, linear regression is used to adjust the coordinates of the proposed regions of interest, refining the precise location of the detected landmarks.

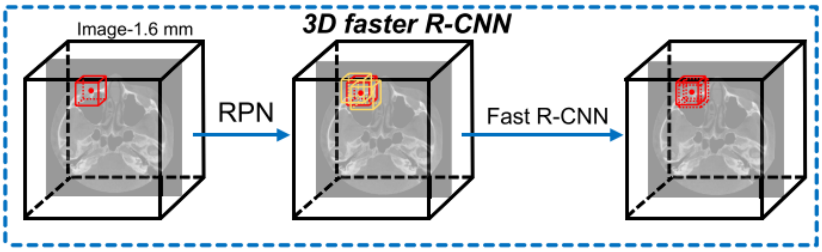


Figure 2.3: 3D faster R-CNN architecture [37].

2.4.4 CNN using point clouds

The scientific paper "Extending Convolutional Pose Machines for Facial Landmark Localization in 3D Point Clouds" [38] addresses the problem of localizing facial landmarks in 3D point clouds, sets of points defined in 3D metric space [39], by using a Convolutional Pose Machines (CPM) architecture to handle the task of localizing facial landmarks in three-dimensional data. The architecture implemented in this study is presented in Figure 2.4.

Initially h_i heatmaps are constructed for each landmark, 1D vectors with size equal to the number of 3D points existing in the point cloud, applying a Gaussian peak at the ground truth landmark location.

In a first step, a PointNet++ architecture [40] is used to process 3D point clouds and extract features relevant to the location of landmarks, without the need to convert the data into volumetric representations or 2D images. This block has as output a set of h_i heatmaps, H_0, which contains an initial indication of the estimated positions of the landmarks.

After extraction the estimated positions of the landmarks are refined using a CPM architecture. The CPM is composed of three stages, where each stage corresponds to a series of convolutions that returns a set of heatmaps h_i with the estimated positions of the landmarks. Stage 1 takes H_0 as input and outputs the set H_1. Stage 2 takes H_1 concatenated with the output of the feature extraction block, H_0, as input and outputs the set H_2. Stage 3 takes H_2 concatenated with the output of the feature extraction block, H_0, as input and outputs the set H_3.

In order to calculate the final predictions of the landmark positions, the three highest value points in each h_i, belonging to H_1, H_2 and H_3, are selected. These three points correspond to the most prominent estimated locations for each specific landmark. After selecting these three points in each heatmap h_i, the i^{th} landmark is calculated as the center of mass of the corresponding three points.

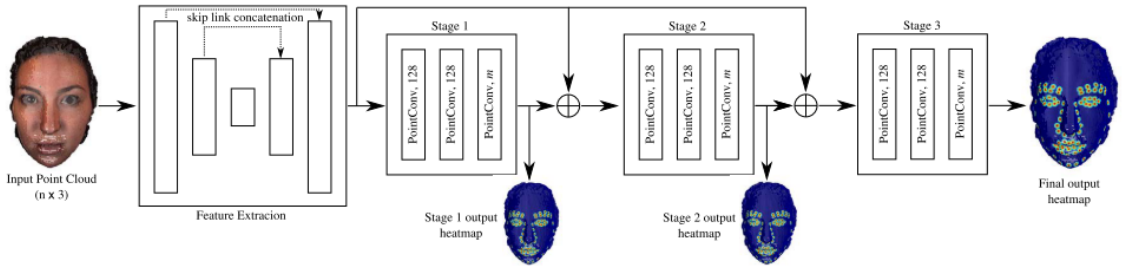


Figure 2.4: Architecture resulting from the combination of PointNet++ and an extension of the CPM architecture [38].

Chapter 3

Data pre-processing and Architectures

This chapter provides a description of the data, where the various types of implants and the corresponding landmarks are presented, the pre-processing of the data for each of the approaches (with respect to checking for valid and invalid implants, as well as conversion and adjustment operations of the 3D representations), and the architecture of the CNNs of each approach.

3.1 Dataset

The dataset consists of meshes (3D representations) of stems, an implant type belonging to the hip implant systems, which were provided by the respective manufacturers to PeekMed®, who subsequently annotated the implants and made them available for use in this study, and a set of Comma-separated values (CSV) files where the information, such as the name and position, of the landmarks corresponding to each implant is stored. Figure 3.1 shows the types of stem shapes present in the dataset.

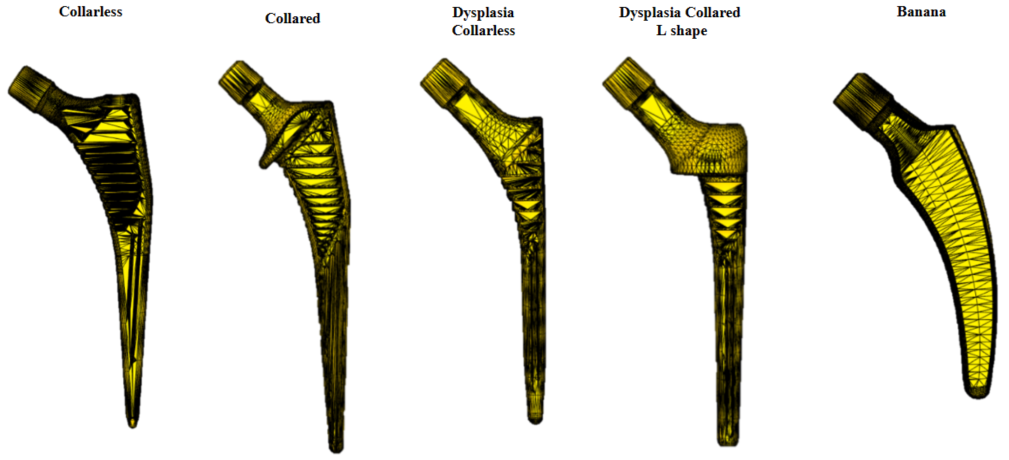


Figure 3.1: Stem shapes.

The available implants belong to the manufacturers: Zimmer Biomet, Atesos and Depuy Synthes. They are divided into a system of parents (in Table 3.1 it is possible to verify the number of implants for each parent), where each parent has several implants that vary according to various properties such as size, side, number of holes, among others. Each landmark is a point P (x, y, z). Table 3.2 shows the possible existing landmarks of an implant (not all implants have the same landmarks, since not all are mandatory).

Parent name	Number of examples
Alloclassic Variall SLV Stem	53
Alloclassic Zweymüller SL Stem	54
Alloclassic Zweymüller SL-64 Stem	6
Alloclassic Zweymüler SL-HAC Stem	46
Alloclassic Zweymüller SLA Stem	24
Alloclassic Zweymüller SLL Revision Stem	34
Alloclassic Zweymüller SLO Stem	56
CORAIL AMT Dysplasia	4
CORAIL AMT Short Neck	24
Fitmore Hip Stem A	56
Fitmore Hip Stem B	53
Fitmore Hip Stem B Extended	53
Fitmore Hip Stem C	56
Pyramid Stem	66
Taperloc 12-14 BM Coated Full Profile	122
Taperloc Micro 12-14 BM Coated	140
Total	847

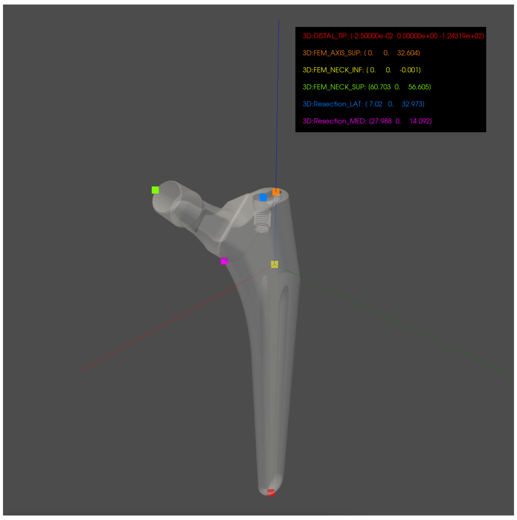
Table 3.1: Amount of implants.

Property	Format
3D:DISTAL_TIP	P (x, y, z)
3D:FEM_AXIS_SUP	P (x, y, z)
3D:Resection_LAT	P (x, y, z)
3D:Resection_MED	P (x, y, z)
3D:FEM_NECK_INF	P (x, y, z)
3D:FEM_NECK_SUP	P (x, y, z)
3D:MED_P	P (x, y, z)
3D:LAT_P	P (x, y, z)

Table 3.2: Implant properties.

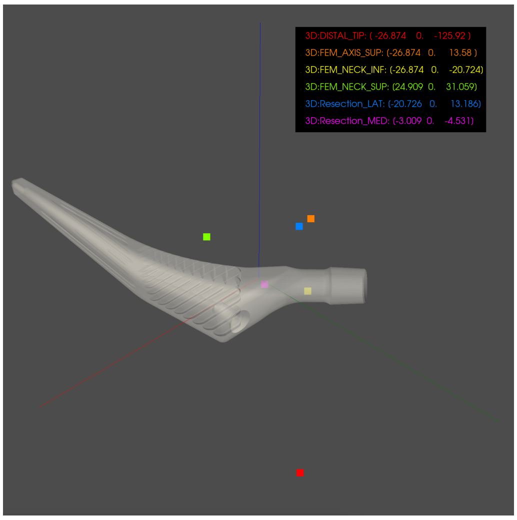
3.2 Data pre-processing

The first step was to analyze the implants and their landmarks. Each implant is a mesh. The names and positions of the landmarks corresponding to each implant are stored in CSV files. To check whether the landmarks corresponding to each implant were correctly positioned, I developed a software that iterates over all implants, presenting each implant and corresponding landmarks, where valid and invalid implants had to be manually selected. Out of a total of 847 implants, 781 valid implants and 66 invalid implants were detected. Figure 3.2 shows an example of a valid implant and Figure 3.3 shows an invalid implant. Of the initial 8 landmarks, only 6 are valid, as the other 2 landmarks have no coordinates in the CSV files. Table 3.3 presents the valid and invalid landmarks.



```
3D:DISTAL_TIP: (-2.50000e-02  0.00000e+00 -1.24319e+02)
3D:FEM_AXIS_SUP: ( 0.     0.    32.604)
3D:FEM_NECK_INF: ( 0.     0.    -0.001)
3D:FEM_NECK_SUP: (60.703  0.    56.605)
3D:Resection_LAT: ( 7.02   0.    32.973)
3D:Resection_MED: (27.988  0.    14.092)
```

Figure 3.2: Valid implant.



```
3D:DISTAL_TIP: ( -26.874   0.    -125.92 )
3D:FEM_AXIS_SUP: (-26.874   0.     13.58 )
3D:FEM_NECK_INF: (-26.874   0.    -20.724)
3D:FEM_NECK_SUP: (24.909  0.    31.059)
3D:Resection_LAT: (-20.726   0.    13.186)
3D:Resection_MED: (-3.009  0.    -4.531)
```

Figure 3.3: Invalid implant.

Valid landmarks	Invalid landmarks
3D:DISTAL_TIP	3D:MED_P
3D:FEM_AXIS_SUP	3D:LAT_P
3D:Resection_LAT	
3D:Resection_MED	
3D:FEM_NECK_INF	
3D:FEM_NECK_SUP	

Table 3.3: Valid and invalid landmarks.

3.2.1 3D volumetric images

First, the original meshes were converted to 3D volumetric images consisting of voxels. After the conversion, it was necessary to scale all the images, so that they all have the same dimensions, as well as their landmarks. The conversion to volumetric images applied a translation transformation to the implant geometry, and it was necessary to correct this problem by applying the same translation transformation to the landmarks corresponding to each volumetric image.

To apply the scaling transformation, a fixed size had to be defined for all images. The defined size was (64 ,32, 128). This size was decided based on the original sizes of each image, which had values close on the axes (x, y, z) to the defined size. Each of the images was scaled, as was its set of landmarks, which were scaled using the same scale factor applied to the corresponding image.

The conversion from meshes to volumetric images caused a displacement of the implant, which led to the need to align the landmarks with the respective implants again. Therefore, after the scale transformation applied to the implants and respective landmarks, a translation transformation had to be applied to the landmarks. By obtaining the volumetric image origin of each implant, a vector was calculated between the original position of each landmark and the origin of the corresponding volumetric image, and this vector was applied to the original position of each landmark. Figure 3.4 shows an example of a pre-processed 3D volumetric implant and its landmarks.

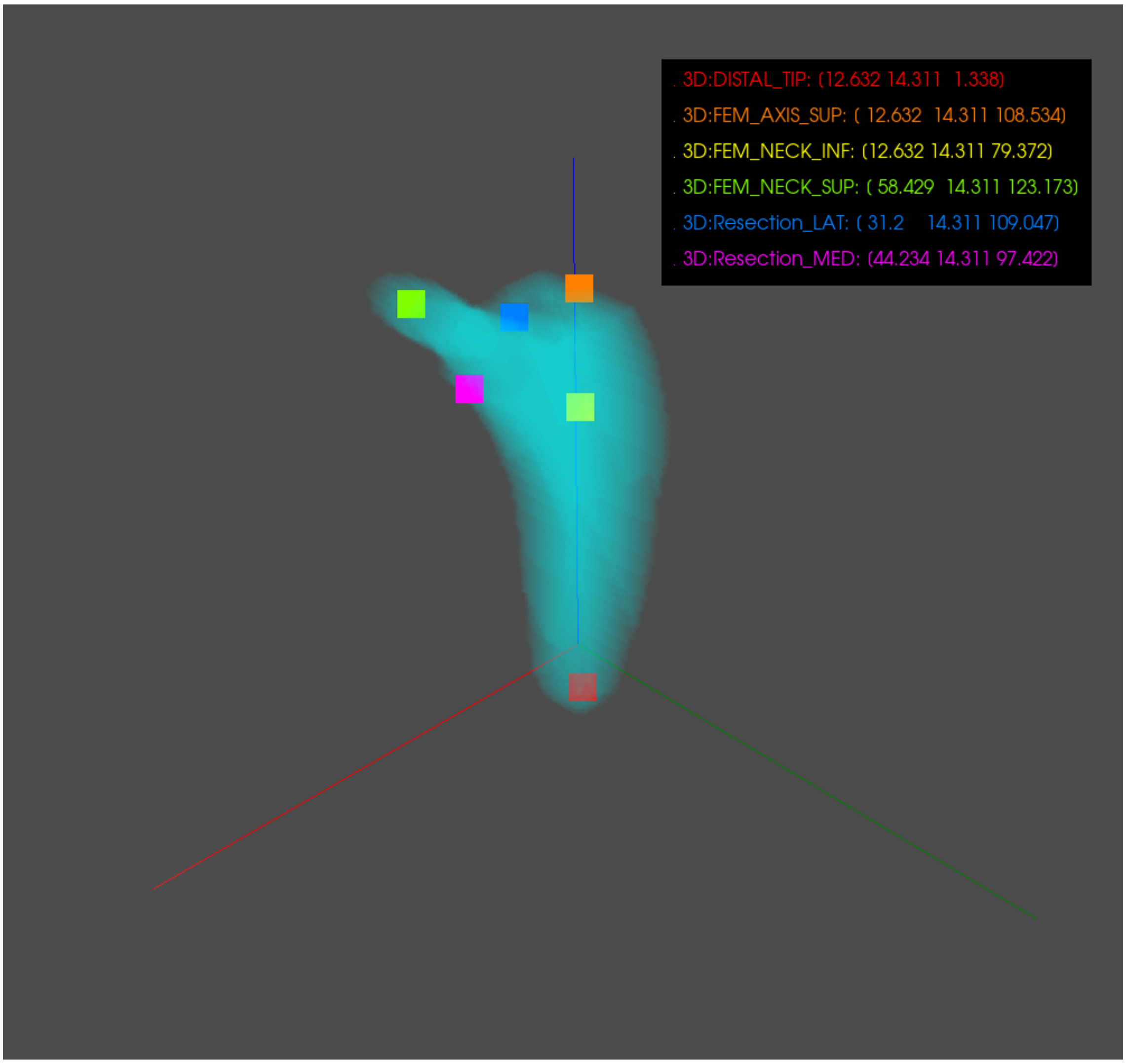

. 3D:DISTAL_TIP: (12.632 14.311 1.338)
. 3D:FEM_AXIS_SUP: (12.632 14.311 108.534)
. 3D:FEM_NECK_INF: (12.632 14.311 79.372)
. 3D:FEM_NECK_SUP: (58.429 14.311 123.173)
. 3D:Resection_LAT: (31.2 14.311 109.047)
. 3D:Resection_MED: (44.234 14.311 97.422)

Figure 3.4: Pre-processed 3D volumetric implant and respective landmarks.

3.2.2 Point cloud

The pre-processing of the meshes in order to obtain point clouds was divided into normalization and centering.

Meshes have a scale property that is a metric for their global scale, which is the length of the diagonal of the bounding box aligned by the mesh axis. Normalization consisted of dividing the coordinates of the vertices of each mesh by the scale of the corresponding mesh, normalizing the values in a range of [-1, 1]. The same operation was applied to the landmarks of each mesh, dividing the coordinates of each landmark by the scale of the corresponding mesh.

The centering was done based on a property of the mesh, the center of mass (calculated based on the spatial distribution of the vertices of the mesh). To center the mesh, a vector was calculated between each mesh vertex and the center of mass of the corresponding mesh, and a translation transformation was performed by applying the corresponding vector to each vertex. As with normalization, the centering operation was applied to the landmarks of each mesh, applying to each landmark the vector of that landmark and the center of mass of the corresponding mesh.

After normalizing and centering the meshes and landmarks, 4096 points were extracted, generating a point cloud for each mesh. Figure 3.5 shows an example of pre-processed point cloud implant and respective landmarks.

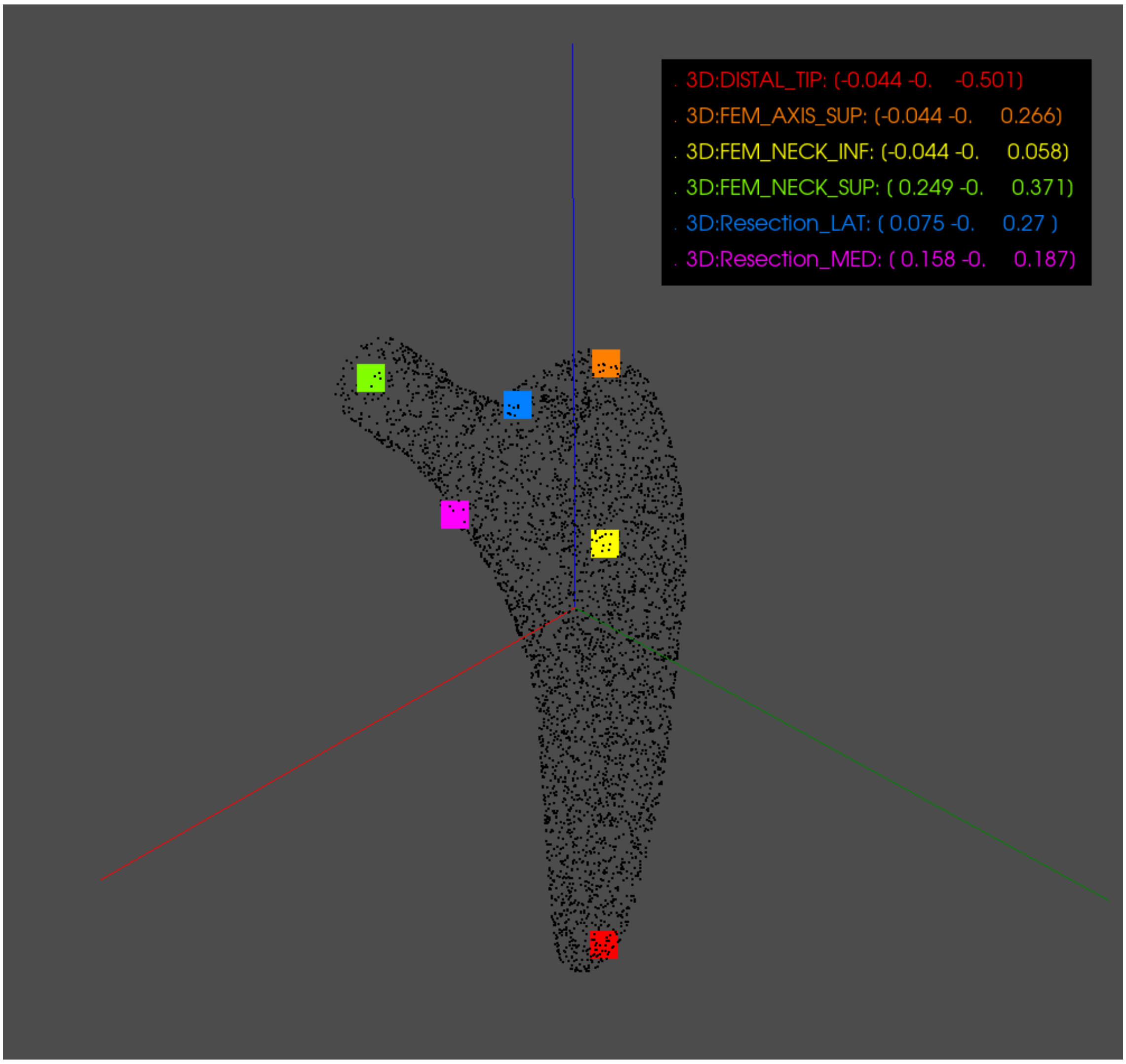

3D:DISTAL_TIP: (-0.044 -0.   -0.501)
3D:FEM_AXIS_SUP: (-0.044 -0.   0.266)
3D:FEM_NECK_INF: (-0.044 -0.   0.058)
3D:FEM_NECK_SUP: (0.249 -0.   0.371)
3D:Resection_LAT: (0.075 -0.   0.27)
3D:Resection_MED: (0.158 -0.   0.187)

Figure 3.5: Pre-processed point cloud implant and respective landmarks.

3.3 Architectures

This section presents the implemented CNN architectures and changes needed to adapt to the problem presented in this thesis, as well as scientific papers where original architectures were presented.

The architectures were implemented in Python (version 3.8.16) using the Keras library (version 2.8.0), a library focused on building and training neural networks. The model trainings were performed on a machine equipped with an NVIDIA A10G graphics processing unit.

3.3.1 Dubost's CNN-inspired architecture

The architecture described in this section is based on the architecture presented in "3D Regression Neural Network for the Quantification of Enlarged Perivascular Spaces in Brain MRI" [41] by Dubost, used to detect objects in enlarged perivascular spaces in brain MRI images. This architecture uses volumetric data as input. To adapt this network to the presented problem, some changes had to be made to the architecture presented in the paper. Figure 3.6 shows the final architecture.

The architecture consists of two blocks composed of four convolution layers and one max pooling layer. The first block is composed of four 3D convolution layers with 32 filters and a kernel size (3*3*3), (Conv, 32, 3*3*3), and a max pooling layer with a pool size (2*2*2), (MaxPool, 2*2*2). The second block consists of four 3D convolution layers with 64 filters and a kernel size (3*3*3), (Conv, 64, 3*3*3), and a max pooling layer with a pool size (2*2*2), (MaxPool, 2*2*2). To ensure that the downsampling was adjusted to the initial size of the volumetric images, the layers (Conv, 128, 3*3*3) and (MaxPool, 4*4*4) were removed. After the two convolution and max pooling blocks, the global average pooling layer and the two fully connected layers, (FC, 2000), from the original architecture were kept. In addition, the original layer, (FC, 1), was converted to (FC, 18), keeping the linear activation, where 18 corresponds to 3 * 6 = 18, where 3 represents the coordinates (x, y, z) of each landmark and 6 represents the number of landmarks.

Finally, after the layer (FC, 18), a reshape of the data was performed, (Reshape, 6, 3), converting the output of the (FC, 18) layer into a matrix of 6 rows, corresponding to the 6 landmarks, and 3 columns, corresponding to the coordinates (x, y, z) of each landmark.

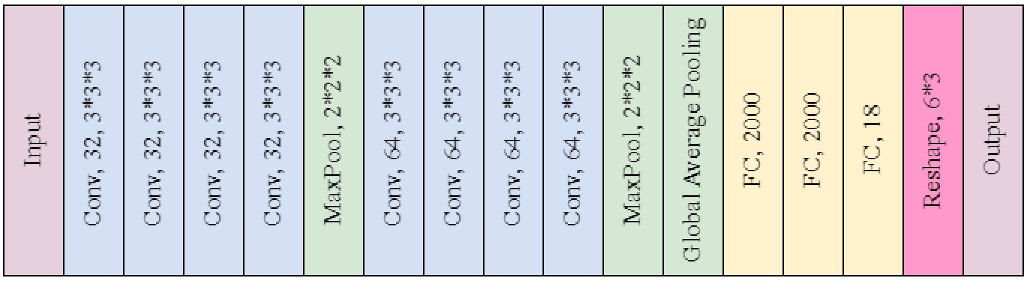


Figure 3.6: Implemented architecture inspired by Dubost's CNN.

3.3.2 ResNet-34

ResNet is an CNN architecture presented in "Deep Residual Learning for Image Recognition" [42], known for its excellent results on classification, segmentation and object detection tasks.

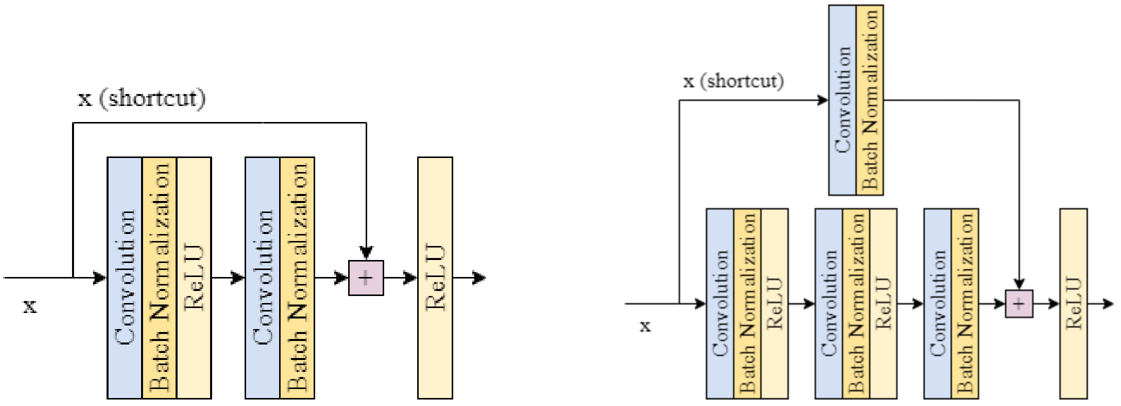
ResNet is short for "Residual Neural Network" and consists of building the network in residual blocks, allowing to create shortcuts between the input layers directly to the output layers, allowing information to be transmitted without significant losses. There are two types of residual blocks: identity blocks and convolution blocks.

The identity block (Figure 3.7a) is used for situations where the dimensions of the input data are the same as the dimensions of the output data. This block consists of a sequence of convolutional layers, batch normalization and ReLU activations. The shortcut connection consists of adding the original input to the output of the block.

The convolution block (Figure 3.7b) changes the dimensions of the data. Similar to the identity block, the convolution block consists of a sequence of convolutional layers, batch normalization and ReLU activations. In addition, it has an additional convolutional layer in the shortcut connection to adjust the dimensions of the input before adding to the output of the block.

ResNet has some variants, such as ResNet-18, ResNet-34, ResNet-50 and others. The name varies according to the resulting sum of the number of convolution and fully connected layers. A ResNet-34 was implemented (Figure 3.8), and it was necessary to change the architecture to adapt to the problem presented.

The last fully connected layer, (FC, 1000), which has a softmax activation, was changed to (FC, 18), with a linear activation, where 18 corresponds to 3 * 6 = 18, where 3 represents the coordinates (x, y, z) of each landmark and 6 corresponds to the total number of landmarks. Finally, a reshape, (Reshape, 6, 3), was performed, which transforms the output of the (FC, 18) layer into a matrix with 6 rows, each row corresponding to a landmark, and 3 columns, corresponding to the coordinates (x, y, z) of each landmark.



(a) Identity block.

(b) Convolution block.

Figure 3.7: Residual blocks.

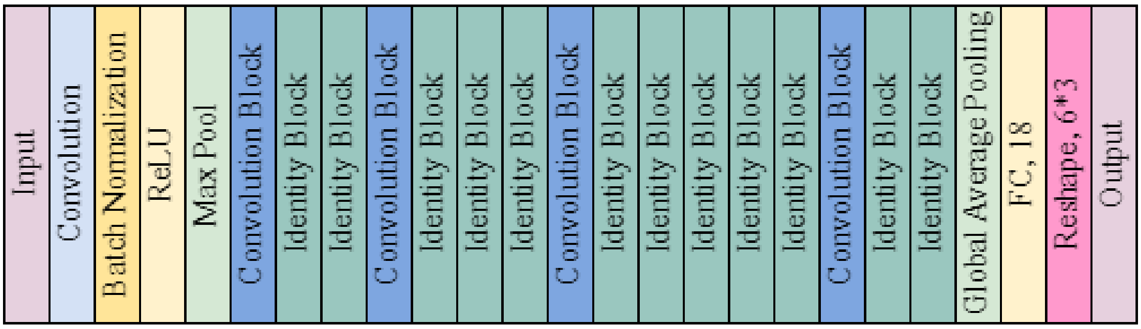


Figure 3.8: ResNet-34 architecture implemented.

3.3.3 PointNet

In this approach, an architecture based on PointNet was implemented. PointNet is a CNN architecture, presented in "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" [43], that works with point clouds in classification and segmentation tasks.

The original PointNet is composed of a classification network and a segmentation network that is an extension of the classification network. For the problem presented, the classification network was implemented, but given that it is a regression problem, some adjustments were made to the architecture.

The PointNet classification network takes n points as input, applying an input transform using a mini-network called a T-net. This first T-net can predict an affine transformation and apply that transformation directly to the coordinates of the input points, helping to make PointNet invariant to global rotations and translations. Next, the points pass through an Multilayer Perceptron (MLP), MLP(64, 64), responsible for capturing local features of the points. The next step is to apply a feature transform using a second T-net, refining the affine transform applied by the first T-net. After the second T-net, the points pass through an MLP(64, 126, 1024) that processes local features of the points. Afterwards a max pooling is applied, aggregating the point features. Finally, the points pass through an MLP(512, 256, k) that processes the aggregated global information, where k corresponds to the number of classes to be classified.

The changes to the architecture consisted of converting the last fully connected layer, (FC, k), which has a softmax activation, to (FC, 18), with a linear activation (since this is a regression problem), where 18 corresponds to 3 * 6 = 18, where 3 corresponds to the coordinates (x, y, z) of each landmark and 6 corresponds to the number of landmarks. Next, a reshape, (Reshape, 6, 3), was performed to convert the output of the (FC, 18) layer into a matrix with 6 rows, corresponding to the 6 landmarks, and 3 columns, corresponding to the coordinates (x, y, z) of each landmark. Figure 3.9 presents architecture implemented.

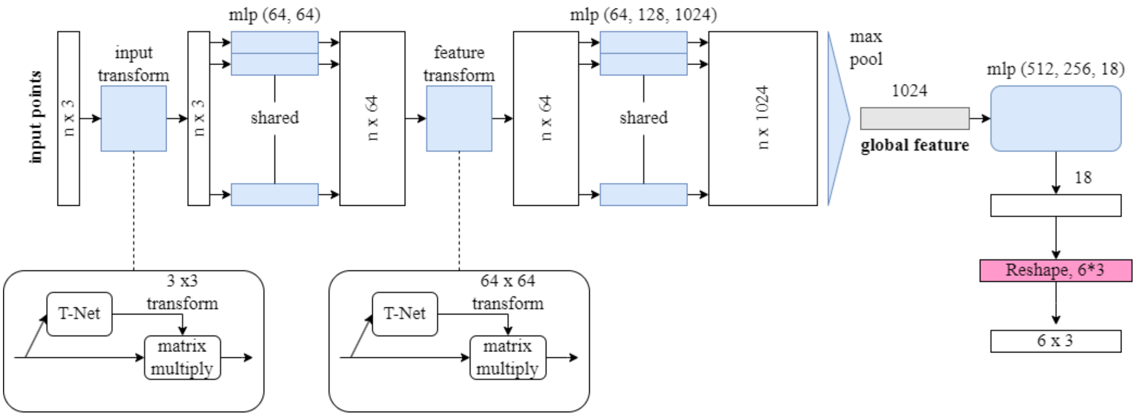


Figure 3.9: PointNet architecture implemented.

This page is intentionally left blank.

Chapter 4

Results and Discussion

4.1 Results

This section presents the results of each of the implemented approaches, a comparison between the original landmarks and those predicted by the models, as well as a description of the training parameters.

The data split (Table 4.1) consisted of 70% data for training, 10% for validation, and 20% for testing. This split was applied for both the implants (volumetric data or point clouds, depending on the model architecture where the data was used as input) and the landmarks corresponding to each implant. All architectures used the Adam optimizer and Mean Square Error (MSE) (Equation 4.1) as cost function.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

For each of the three architectures implemented, several experiments were performed, varying the number of epochs and the learning rate. This allowed for an analysis of evaluation metrics, such as graphs of the evolution of training and validation losses. As a metric for evaluating the performance of the models, the Root Mean Square Error (RMSE) was calculated, on the test dataset, between the original and predicted landmarks. The RMSE was used instead of MSE so that it is possible to compare the distances in *mm*, rather than in *mm*^{2}.

In data pre-processing, the landmarks were subjected to scale and translation transformations for both the volumetric data and the point clouds. The landmarks predicted by the models were readjusted, applying the inverse scale and translation transformations, in order to be able to compare with the original landmarks by calculating the RMSE.

Group	Percentage (%)	Total
Train	70	546
Validation	10	78
Test	20	157
		781

Table 4.1: Data split.

4.1.1 Dubost's CNN-inspired architecture

The initial experiments were performed with a learning rate of 1×10^{-3}, but none of them showed a significant decrease in the validation loss curve, generating oscillations in the curve. Then the learning rate was reduced to 1×10^{-4}, which led to a gradual decrease in the validation loss, always close to the training loss, showing a good fit for the model. The learning rate was then decreased to 1×10^{-5}, where the validation loss curve showed an initial decrease, but quickly started to rise in the initial epochs.

The next experiments consisted of varying the number of epochs up to a maximum of 100. In all experiments, the validation loss curve showed an initial rise and oscillated throughout training, always remaining above the training loss curve. Based on this analysis, it was decided to add the early stopping mechanism to avoid unnecessary training of the model.

Based on the evolution of the loss curves, the number of tolerance epochs and the minimum improvement value for early stopping were defined. Using early stopping, training was performed with 100 epochs, varying the number of tolerance epochs for improvement and the minimum improvement value.

The experiment that obtained the best performance was the one with 100 epochs, 12 rounds of tolerance for improvement, and a minimum improvement value of 6 for the validation loss, where training stopped at epoch 18. This experiment resulted in a RMSE of $2.898mm \pm 0.969mm$. It is possible to observe the results of this experiment in Table 4.2, which shows the RMSE between the original landmarks and those predicted by the model, Figure 4.1 presents the evolution of the loss curves, Figure 4.2 shows the best prediction, Figure 4.3 shows the worst prediction and Figure 4.4 shows a random prediction.

Landmark	RMSE (mm)
3D:DISTAL_TIP	1.765 ± 1.02
3D:FEM_AXIS_SUP	2.6 ± 1.381
3D:FEM_NECK_INF	4.083 ± 2.186
3D:FEM_NECK_SUP	4.369 ± 2.953
3D:Resection_LAT	2.766 ± 1.748
3D:Resection_MED	3.056 ± 1.596

Table 4.2: RMSE between the original and the predicted landmarks, from test set, by Dubost's CNN-inspired model.

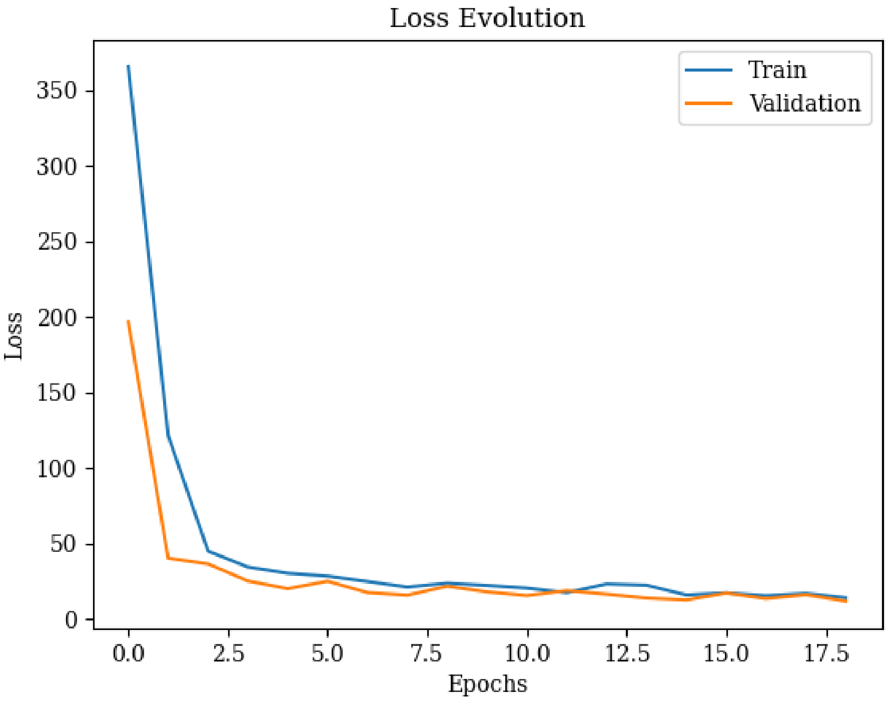


Figure 4.1: Dubost's CNN-inspired model loss curves evolution.

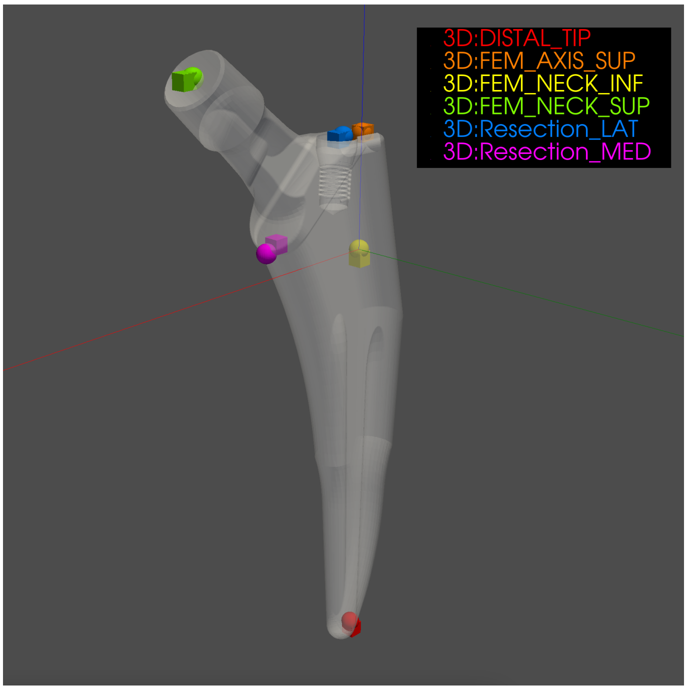


Figure 4.2: Dubost's CNN-inspired model best prediction. RMSE: $0.96mm \pm 0.438mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

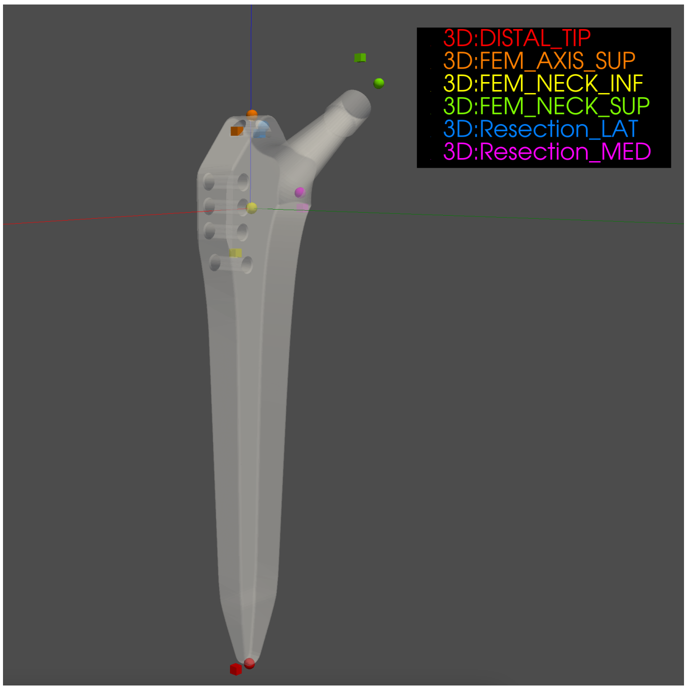


Figure 4.3: Dubost's CNN-inspired model worst prediction. RMSE: $5.762mm \pm 2.893mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

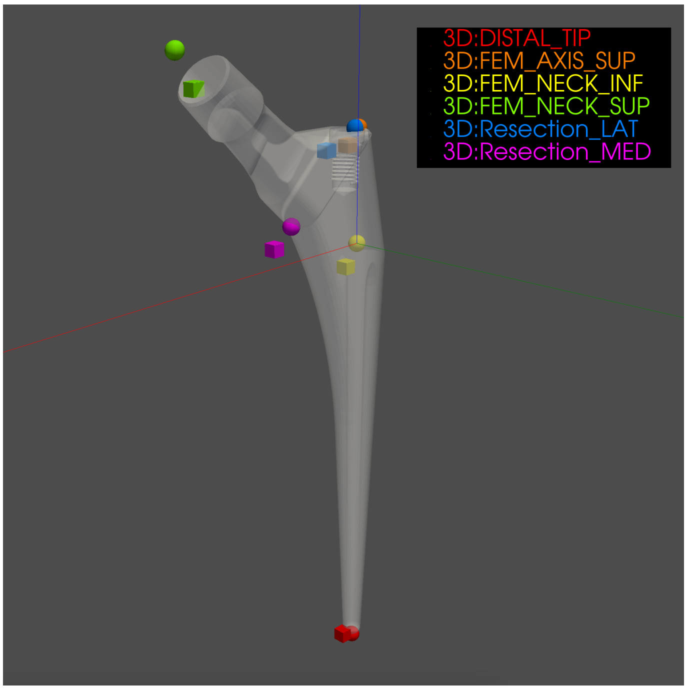


Figure 4.4: Dubost's CNN-inspired model random prediction. RMSE: $3.932mm \pm 1.369mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

4.1.2 ResNet-34

The first experiments showed that a high learning rate resulted in an inconsistent evolution of the validation loss curve, resulting in oscillations in that curve. By decreasing the learning rate to 1×10^{-5}, the validation loss curve was gradually reduced, approaching the training loss curve, showing that the learning rate of 1×10^{-5} fitted the model well.

Fixing the learning rate at 1×10^{-5}, several experiments were performed varying the number of epochs. An experiment with 40 epochs resulted in a mean error of $7.733mm \pm 3.662mm$. It was decided to increase the number of epochs, until an experiment with 100 epochs achieved even lower mean error values, with $7.452mm \pm 3.38mm$, similar to the 40 epoch experiment.

In addition to these results, an analysis of the evolution of the loss curves showed that they stagnated after a certain point. To deal with this, the early stopping mechanism was added. The determination of the optimal number of tolerance epochs and the minimum improvement threshold for early stopping was based on an analysis of the loss curves evolution.

Using early stopping, a training of 100 epochs was performed, varying the number of tolerance epochs and the minimum improvement value. The experiment that obtained the best performance was the one with 100 epochs that stopped at epoch 43, 6 rounds of tolerance for improvement, and a minimum improvement value of 12 for the validation loss. This experiment resulted in a RMSE of $7.112mm \pm 4.08mm$, lower than the RMSE of the training runs without early stopping. The outcomes of this experiment can be observed in Table 4.3, which presents the RMSE between the original landmarks and the predicted landmarks by the model. The progression of the loss curves is illustrated in Figure 4.5 Additionally, Figures 4.6, 4.7 and 4.8 showcase the best prediction, the worst, and a randomly selected, respectively.

Landmark	RMSE (mm)
3D:DISTAL_TIP	4.259 ± 3.075
3D:FEM_AXIS_SUP	9.378 ± 6.128
3D:FEM_NECK_INF	7.73 ± 5.229
3D:FEM_NECK_SUP	11.54 ± 6.322
3D:Resection_LAT	9.124 ± 5.468
3D:Resection_MED	7.559 ± 5.389

Table 4.3: RMSE between the original and the predicted landmarks, from test set, by ResNet-34 model.

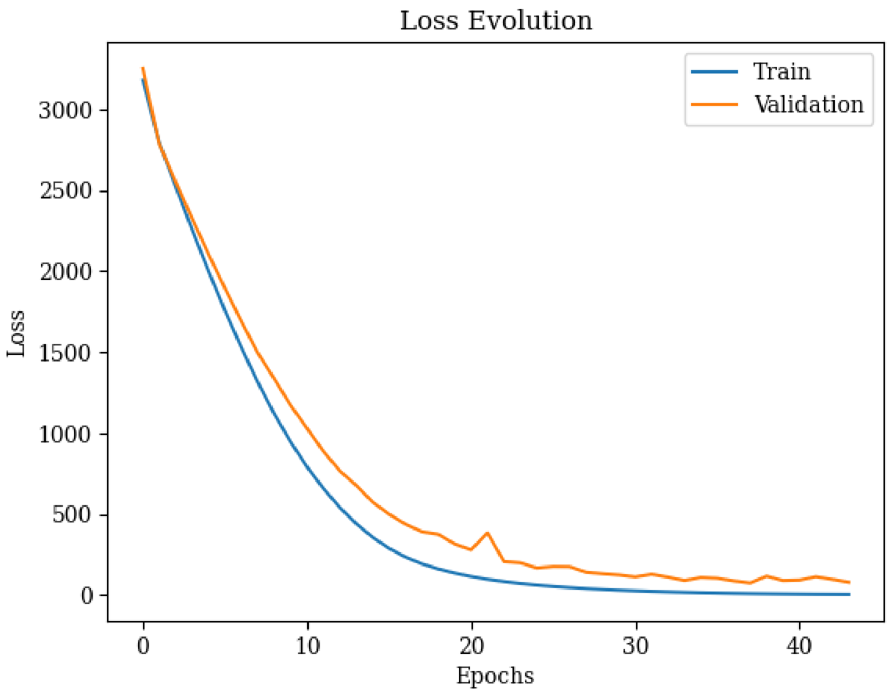


Figure 4.5: ResNet-34 model loss curves evolution.

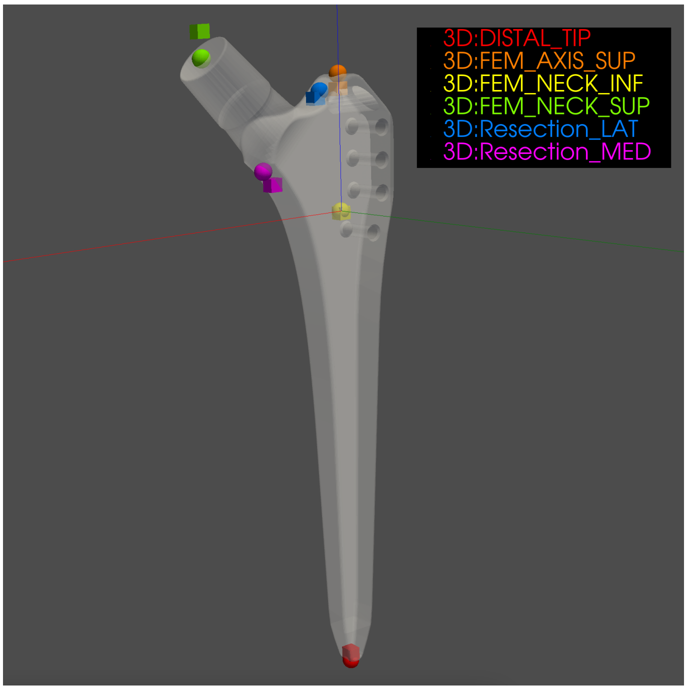


Figure 4.6: ResNet-34 model best prediction. RMSE: $1.636mm \pm 0.967mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

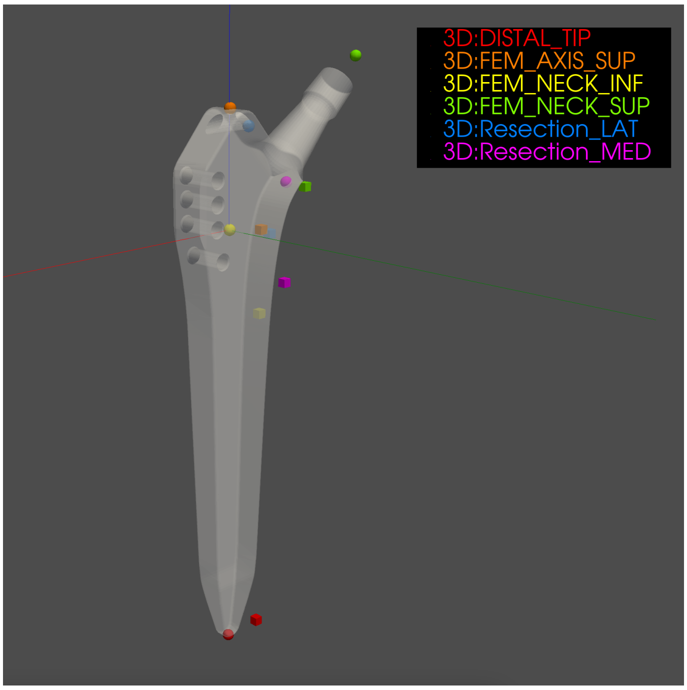


Figure 4.7: ResNet-34 model worst prediction. RMSE: 21.424$mm \pm$ 6.046mm. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

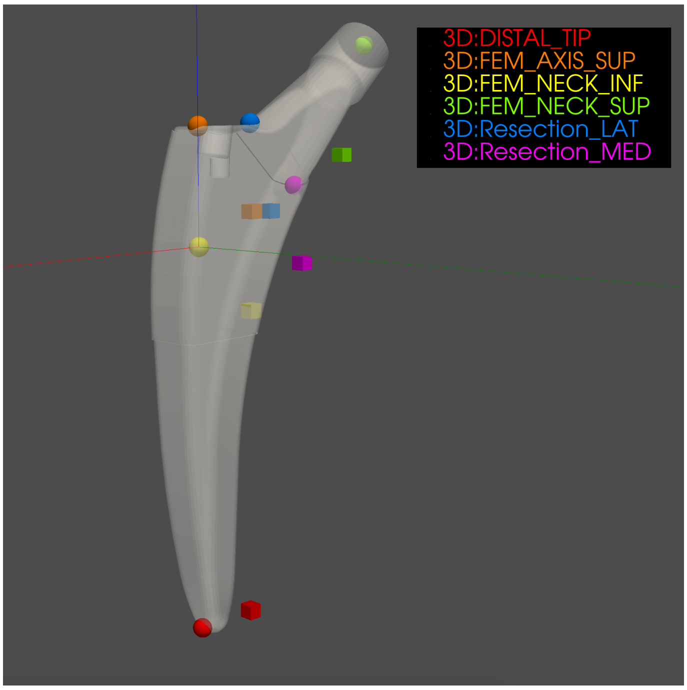


Figure 4.8: ResNet-34 model random prediction. RMSE: 11.764$mm \pm$ 1.815mm. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

4.1.3 PointNet

Experiments were performed to try to understand the evolution of the loss curves. Several learning rates were tested, and lower learning rates were preferred, as they resulted in smoothly descending curves, avoiding unwanted oscillations, keeping the curves close to each other, verifying that a learning rate of 1×10^{-4} was the most adequate for the model.

Fixing the learning rate at 1×10^{-4}, several experiments were performed, varying the number of epochs. It was observed that increasing the number of epochs resulted in a gradual improvement of the results. Of the experiments performed, training with 100 epochs showed the best results, with a RMSE of $0.895mm \pm 0.539mm$.

However, when performing a test with 200 epochs, it was found that the results were worse compared to the test with 100 epochs. Analysis of the loss curves revealed that after a certain point the curves started to stagnate. To deal with this problem, early stopping was implemented. Through the analysis of the evolution values of the loss curves, a number of epochs of tolerance and the minimum desired improvement value was defined. With the implementation of early stopping, 200 epochs of training were performed, varying the tolerance and the minimum improvement value.

With a tolerance of 12 epochs and a minimum improvement value of 1×10^{-11} for the validation loss, a training of 200 epochs stopped at epoch 127, with a RMSE of $0.857mm \pm 0.452mm$, lower than the RMSE of the training of 100 epochs without early stopping. Table 4.4 shows the RMSE between the original landmarks and those predicted by this model. Figure 4.9 presents the evolution of the loss curves. Figures 4.10, 4.11, 4.12 present the best prediction, the worst and a random one, respectively.

Landmark	RMSE (mm)
3D:DISTAL_TIP	0.703 ± 0.419
3D:FEM_AXIS_SUP	0.645 ± 0.401
3D:FEM_NECK_INF	0.988 ± 1.023
3D:FEM_NECK_SUP	1.433 ± 1.093
3D:Resection_LAT	0.835 ± 0.475
3D:Resection_MED	0.941 ± 0.665

Table 4.4: RMSE between the original and the predicted landmarks, from test set, by PointNet model.

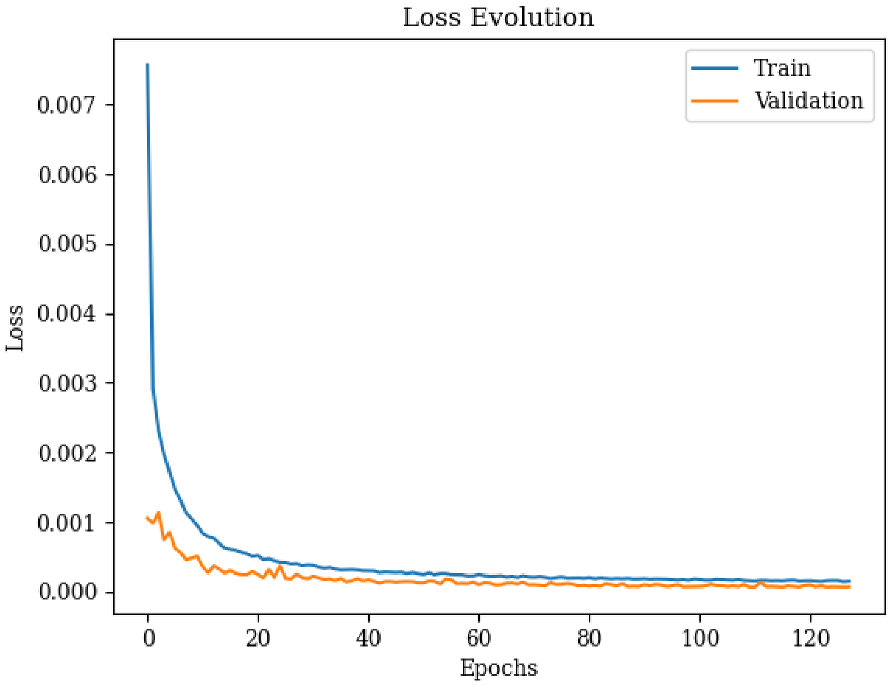


Figure 4.9: PointNet model loss curves evolution.

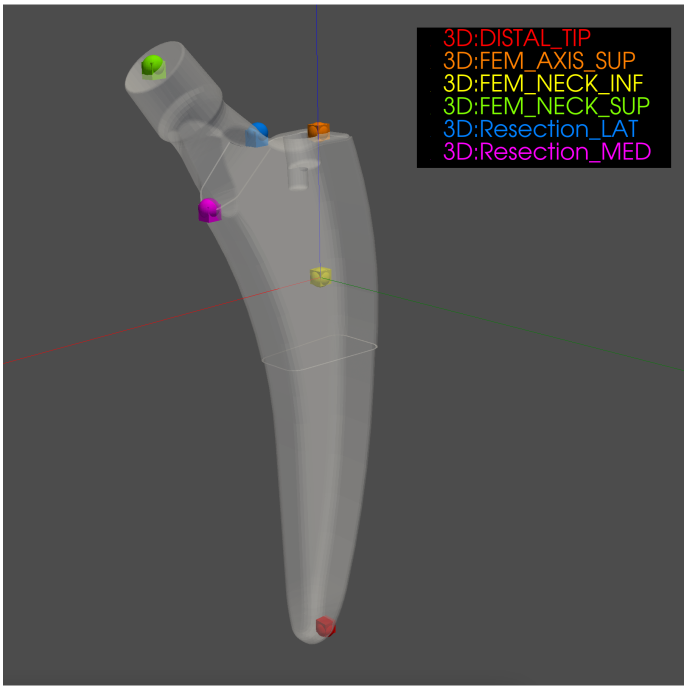


Figure 4.10: PointNet model best prediction. RMSE: $0.299mm \pm 0.151mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

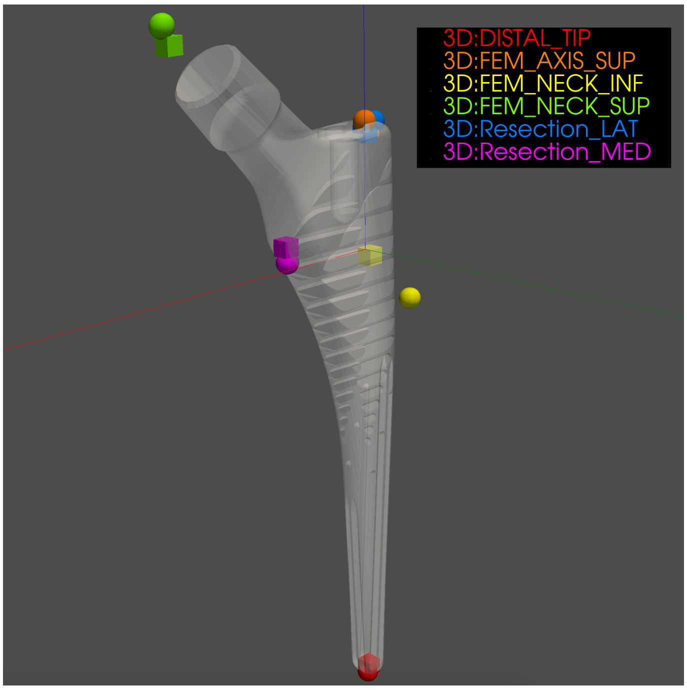


Figure 4.11: PointNet model worst prediction. RMSE: $2.54mm \pm 2.722mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

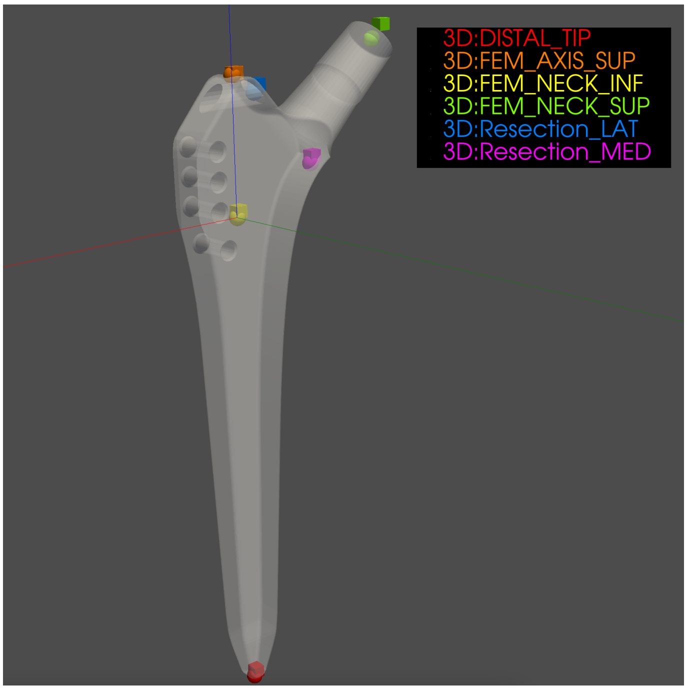


Figure 4.12: PointNet model random prediction. RMSE: $1.125mm \pm 0.657mm$. The spheres represent the original landmarks and the cubes represent the predicted landmarks.

4.2 Discussion

Model	RMSE (*mm*)
Dubost's CNN-inspired	2.898 ± 0.969
ResNet-34	7.112 ± 4.08
PointNet	0.857 ± 0.452

Table 4.5: RMSE comparison between models

Stems can have different lengths, as is presented in the scientific article "The impact of reducing the femoral stem length in total hip arthroplasty during gait" [44], which presents plannings of Total Hip Arthroplasty (THA) where stems of different lengths were used, such as 100*mm* and 140-166*mm*. With these values as reference, the results of each of the models presented in this thesis will be presented and discussed. Table 4.5 presents the RMSE, across all original and predicted landmarks, for each model.

The model inspired by Dubost's CNN obtained a RMSE of $2.898mm \pm 0.969mm$, a considerable error with respect to the average stem length. Regarding training, it was complicated to find suitable training parameters, namely learning rate, number of epochs, as well as the parameters for early stopping (number of tolerance epochs and minimum validation loss improvement value). The training and validation loss curves (Figure 4.1) presented a descending evolution, reaching a point of stability, but suffering several oscillations, with the curves crossing each other, possibly indicating the presence of overfitting. To improve the training of this model, it would be interesting to combine other training parameters, explore regularization techniques such as dropout to try to reduce overfitting and improve the generalization ability of the model, as well as a larger dataset or data augmentation techniques.

ResNet-34 presented an RMSE of $7.112mm \pm 4.08mm$, an excessive error considering the average stem length, which can compromise the patient's health. Regarding training, both loss curves (Figure 4.5) showed a descending evolution, with a reduced gap between them. The validation loss curve always remained above the training loss curve, which may indicate overfitting. This may result from inadequate generalization by the model, caused by an overfitting to the training data, as well as insufficient training data or inadequate training parameters. Therefore, as improvement techniques, it would be indicated to increase the amount of training data, regularization techniques and adjustment of training parameters.

PointNet presented an RMSE of $0.857mm \pm 0.452mm$, an acceptable error considering the average stem length. The training and validation loss curves (Figure 4.9) showed a descending evolution with a reduced gap between the curves, where the validation loss curve was always below the training loss curve. This indicates a good generalization ability for data not seen during training. Still, it would be possible to improve the performance of this architecture by applying regularization techniques to improve the generalization ability of the model, as well as a larger dataset.

At the dataset level, the shapes of the stems were very similar, apart from the banana-shaped stems, which are a bit different from the rest. Similarities between the shapes of the stems may make the task of landmark detection easier, as the models can learn patterns and features common to all the implants. Even with the similarities between the implants, Dubost's CNN-inspired model and ResNet-34 showed unsatisfactory results compared to PointNet.

PointNet is specifically designed to process point clouds, which represent the surface of objects in 3D and is able to capture spatial and relational information of the points, including the geometry and arrangement of landmarks on the implant surface. On the other hand, Dubost's CNN-inspired model and ResNet-34 process volumetric data that has additional information, such as the internal structure of the implant, which increases the dimensionality of the data, which can lead to a higher amount of noise and redundancy in the data, increasing the complexity in the landmark detection task. This may be one of the reasons for the better performance of PointNet compared to the other architectures.

Overall, all three architectures proved to be able to detect landmarks in orthopedic implants, with PointNet being the most promising. With proper adjustments and regularization techniques and a larger dataset, all architectures could improve their results.

Chapter 5

Conclusion

In this thesis, three solutions capable of automatically detecting landmarks in orthopedic implants were presented. The implemented approaches consist of CNNs inspired by scientific papers where anatomical landmark detection mechanisms were developed. Although the great expectations about the DL application in this problem, only one CNN architecture, PointNet, showed promising results in some cases.

Two of the architectures, Dubost's CNN inspired model and ResNet-34, which used volumetric data as input, were able to detect landmarks in the stems, but showed considerable errors between the original and predicted landmarks, leading to incorrect implant placement, which could compromise the patient's health. The last architecture, PointNet, was the architecture that required operations at the pre-processing level and was the easiest to find suitable training parameters. It also showed the smallest error between the original and predicted landmarks, a relatively low error considering the average size of a stem. These results have significant implications, as they demonstrate that it is possible to speed up the time-consuming process of landmarking implants, which is usually performed manually by experts, saving time and resources.

As limitations of this study, it was difficult to tune the training parameters of the architectures that process volumetric data, Dubost's CNN-inspired model and ResNet-34, which still obtained poor results. A limitation common to all approaches was the small dataset. With a larger dataset, it would be possible to obtain a more comprehensive representation of the variability of implants, which could lead to better results for all the approaches evaluated, including PointNet, allowing for better generalization and learning ability of the models by providing more examples and variations for training.

For future work, one suggestion would be to test the implemented architectures with a larger dataset, containing more implants of stems with different sizes and shapes, in order to improve their generalization ability. Another suggestion, would be to apply transfer learning techniques, where pre-trained models on similar problems are adapted for the detection of landmarks on implants, could help to leverage prior knowledge and speed up the training process. A third suggestion would be to modify the implemented architectures, since they have been shown to perform this task, small changes in the architecture (such as adding more convolution or pooling layers) could improve the performance of the models.

References

[1] Renaud Lafage, Sébastien Pesenti, Virginie Lafage, and Frank J. Schwab. Self-learning computers for surgical planning and prediction of postoperative alignment. *European Spine Journal*, 27:123–128, 2 2018.

[2] Germano Gomes, Sofie Van Cauter, Matthieu De Beule, Lara Vigneron, Christophe Pattyn, and Emmanuel Audenaert. Patient-specific modelling in orthopedics: From image to surgery, 2013.

[3] Roland Glenister and Sandeep Sharma. Anatomy, bony pelvis and lower limb, hip. *StatPearls*, 7 2022.

[4] S Jorge Siopack, Harry E Jergesen, and San Francisco. Total hip arthroplasty. *Western Journal of Medicine*, 162:243–249, 1995.

[5] Matthew Varacallo, T. David Luo, and Norman A. Johanson. Total hip arthroplasty techniques. *StatPearls*, 9 2022.

[6] Ahmed Siddiqi, Brett R. Levine, and Bryan D. Springer. Highlights of the 2021 american joint replacement registry annual report. *Arthroplasty Today*, 13:205, 2 2022.

[7] Liyao Guo, Seyed Ataollah Naghavi, Ziqiang Wang, Swastina Nath Varma, Zhiwu Han, Zhongwen Yao, Ling Wang, Liqiang Wang, and Chaozong Liu. On the design evolution of hip implants: A review. *Materials & Design*, 216:110552, 4 2022.

[8] Fortune Business Insights. Hip replacement market size, share industry trends [2028]. *Available at: https://www.fortunebusinessinsights.com/industry-reports/hip-replacement-implants-market-100247*, 2021.

[9] James D. Wylie and Robert Z. Tashjian. Planning software and patient-specific instruments in shoulder arthroplasty. *Current Reviews in Musculoskeletal Medicine*, 9:1–9, 3 2016.

[10] Jesús Montúfar, Marcelo Romero, and Rogelio J. Scougall-Vilchis. Hybrid approach for automatic cephalometric landmark annotation on cone-beam computed tomography volumes. *American Journal of Orthodontics and Dentofacial Orthopedics*, 154:140–150, 7 2018.

[11] Florin C. Ghesu, Bogdan Georgescu, Tommaso Mansi, Dominik Neumann, Joachim Hornegger, and Dorin Comaniciu. An artificial agent for anatomical landmark detection in medical images. *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9902 LNCS:229–237, 2016.

[12] Schilders. Hip anatomy. *Available at: http://www.ernestschilders.com/hip-anatomy.php*, 2014.

[13] OrthoInfo-AAOS. *Available at: https://orthoinfo.aaos.org/en/treatment/total-hip-replacement/*, 2020.

[14] Richard Iorio, Jodi Siegel, Lawrence M. Specht, John F. Tilzey, Audrey Hartman, and William L. Healy. A comparison of acetate vs digital templating for preoperative planning of total hip arthroplasty: is digital templating accurate and safe? *The Journal of arthroplasty*, 24:175–179, 2 2009.

[15] Leo Joskowicz. Computer-aided surgery meets predictive, preventive, and personalized medicine. *EPMA Journal*, 8:1–4, 2017.

[16] Jonathan L. Knight and Richard D. Atwater. Preoperative planning for total hip arthroplasty. quantitating its utility and precision. *The Journal of arthroplasty*, 7 Suppl:403–409, 1992.

[17] Andrew R. Hsu, Jeffrey D. Kim, Sanjeev Bhatia, and Brett R. Levine. Effect of training level on accuracy of digital templating in primary total hip and knee arthroplasty. *Orthopedics*, 35, 2 2012.

[18] Alejandro González Della Valle, Douglas E. Padgett, and Eduardo A. Salvati. Preoperative planning for primary total hip arthroplasty. *The Journal of the American Academy of Orthopaedic Surgeons*, 13:455–462, 2005.

[19] Shoaleh Shahidi, Ehsan Bahrampour, Elham Soltanimehr, Ali Zamani, Morteza Oshagh, Marzieh Moattari, and Alireza Mehdizadeh. The accuracy of a designed software for automated localization of craniofacial landmarks on cbct images. *BMC Medical Imaging*, 14, 9 2014.

[20] Kaiwei Che, Chengwei Ye, Yibing Yao, Nachuan Ma, Ruo Zhang, Jiankun Wang, and Max Q. H. Meng. Deep learning-based biological anatomical landmark detection in colonoscopy videos. *arXiv (preprint)*, 8 2021.

[21] Dong Yang, Shaoting Zhang, Zhennan Yan, Chaowei Tan, Kang Li, and Dimitris Metaxas. Automated anatomical landmark detection ondistal femur surface using convolutional neural network. *IEEE 12th international symposium on biomedical imaging (ISBI)*, 2015-July:17–21, 7 2015.

[22] G. Dave Singh and Manarshhjot Singh. Virtual surgical planning: Modeling from the present to the future. *Journal of Clinical Medicine*, 10, 12 2021.

[23] A D Levy-Mandel, A N Venetsanopoulos, and J K Tsotsos. Knowledge-based landmarking of cephalograms. *Computers and Biomedical Research*, 19:282–309, 1986.

[24] Ramakant Nevatia. Evaluation of a simplified hueckel edge-line detector. *Computer Graphics and Image Processing*, 6:582–588, 12 1977.

[25] Jan Ehrhardt, Heinz Handels, Bernd Strathmann, Thomas Malina, Werner Plötz, and Siegfried J. Pöppl. Atlas-based recognition of anatomical structures and landmarks to support the virtual three-dimensional planning of hip operations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2878:17–24, 2003.

[26] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21:347–353, 1987.

[27] Hye Sun Yun, Tae Jun Jang, Sung Min Lee, Sang Hwy Lee, and Jin Keun Seo. Learning-based local-to-global landmark annotation for automatic 3d cephalometry. *Physics in Medicine and Biology*, 65, 4 2020.

[28] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111:257–276, 5 2019.

[29] Yali Li, Shengjin Wang, Qi Tian, and Xiaoqing Ding. Feature representation for statistical-learning-based object detection: A review. *Pattern Recognition*, 48:3542–3559, 11 2015.

[30] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Rajiv Suman, and Shanay Rab. Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, 3:58–73, 1 2022.

[31] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv (preprint)*, 11 2015.

[32] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J Big Data*, 8:53, 2021.

[33] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[34] Satya P Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan, and Balázs Gulyás. 3d deep learning on medical images: A review. *Sensors*, 20, 2020.

[35] Daniel Filipe, Gonçalves Cruz, João Pedro, and Almeida Barreto. Automatic anatomical landmark location estimation in orthopedics, 2021.

[36] Zhe Zhang, Jie Tang, and Gangshan Wu. Simple and lightweight human pose estimation. 11 2019.

[37] Xiaoyang Chen, Chunfeng Lian, Hannah H. Deng, Tianshu Kuang, Hung Ying Lin, Deqiang Xiao, Jaime Gateno, Dinggang Shen, James J. Xia, and Pew Thian Yap. Fast and accurate craniomaxillofacial landmark detection via 3d faster r-cnn. *IEEE Transactions on Medical Imaging*, 40:3867–3878, 12 2021.

[38] Eimear O' Sullivan and Stefanos Zafeiriou. Extending convolutional pose machines for facial landmark localization in 3d point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[39] Saifullahi Aminu Bello, Shangshu Yu, and Cheng Wang. Review: deep learning on 3d point clouds. *Remote Sensing*, 12, 1 2020.

[40] Charles R Qi Li, Yi Hao, Su Leonidas, and J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[41] Florian Dubost, Hieab Adams, Gerda Bortsova, M Arfan Ikram, Wiro Niessen, Meike Vernooij, and Marleen De Bruijne. 3d regression neural network for the quantification of enlarged perivascular spaces in brain mri. *Medical image analysis*, 51:89–100, 2019.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE conference on computer vision and pattern recognition*, pages 770–778, 12 2015.

[43] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 12 2016.

[44] Anatole Vilhelm Wiik, Adeel Aqil, Bilal Al-Obaidi, Mads Brevadt, and Justin Peter Cobb. The impact of reducing the femoral stem length in total hip arthroplasty during gait. *Archives of Orthopaedic and Trauma Surgery*, 141:1993–2000, 11 2021.