Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING

Master of Science in Biomedical Engineering

CEL-Unet: a novel CNN architecture for 3D Segmentation of Knee Bones affected by Severe Osteoarthritis for PSI-Based Surgical Planning

Supervisor

Prof. Pietro Cerveri

Co-Supervisor

Ing. Davide Marzorati

Candidate

Alberto Faglia

Student ID: 921271

Academic Year 2020 – 2021

Ringraziamenti

Vorrei ringraziare tutte le persone che mi hanno sostenuto nello svolgimento di questo lungo percorso, e che ne hanno permesso la piena realizzazione. Sono tante, e ne sono immensamente felice.

Ringrazio il Politecnico di Milano, per avermi accolto in grande armonia, per il sapere che mi ha trasmesso e per il senso di appartenenza che mi ha lasciato e che terrò dentro per sempre.

Ringrazio il mio relatore Prof. Pietro Cerveri e il mio correlatore Ing. Davide Marzorati, per avermi dato l'opportunità di approfondire un argomento di così grande interesse scientifico e umano, supportandomi e indirizzandomi costantemente nel miglior modo possibile verso il completamento di questo lavoro.

Ringrazio tutti i coinquilini passati in "Casa Fabogneso", che hanno rappresentato la mia seconda casa durante questi anni di università, e con i quali si è creato un legame speciale, esattamente come quello che nasce in una magnifica e dinamica famiglia.

Ringrazio dal profondo Schüttstraße 1.2.1 insieme a tutti gli amici e le amiche che hanno reso indimenticabili i 5 mesi passati ad Amburgo. Harburg still rules.

Ringrazio di cuore gli amici e le amiche de "L'UPIESSE", corriere portatore di quotidiana e incondizionata allegria nelle giornate e nelle serate milanesi, che hanno scandito la mia vita di questi anni. Senza di voi, tutto sarebbe stato più difficile.

Ringrazio i "Barriga", per avermi dato e per darmi tuttora un valido motivo per mettere in costante discussione il mio futuro da ingegnere biomedico. Siamo la colonna sonora della mia vita e siamo un sogno che si avvera.

Ringrazio i fratelli della "Cumpa United", per le infinite discussioni, per le partite del sabato, ma soprattutto per tutto ciò che ci sta intorno e per la facilità con cui ogni situazione possa sempre scivolare in un momento memorabile. Sosteniamola fino al cinquantesimo.

Ringrazio immensamente i "Gnari di Colle", con cui ho condiviso la mia vita intera, per la spensieratezza e la giusta pazzia che mi hanno sempre insegnato e per ricordarmi costantemente dove siano le mie radici, anche quando le scelte o le necessità mi portano lontano.

Ringrazio infinitamente tutta la mia famiglia, per avermi sempre e incondizionatamente supportato nelle mie scelte, per essere stata costante fonte di stimoli e per avermi regalato una vita dalla quale non avrei potuto chiedere di più. Grazie alle nonne e ai nonni, che con la loro grande saggezza e interminable disponibilità, sanno sempre come sostenerci nel migliore dei modi.

Infine, ringrazio con immenso amore la mia ragazza, Claudia, che in tutti questi anni ha saputo conoscermi nel profondo, comprendermi nei momenti più bui e supportarmi con grande amore e affetto in tutte le mie esperienze. La forza e determinazione che hai sempre saputo dimostrare e con cui vivi ogni giorno la tua vita, sono state costante fonte di energia e ispirazione per me e per le mie scelte.

Sommario

La medicina moderna si sta sviluppando parallelamente alle nuove tecnologie e soluzioni innovative basate sui dati stanno acquisendo interesse ed efficacia come mai prima d'ora. Le immagini diagnostiche costituiscono una fonte di informazioni cruciale per la pianificazione degli interventi, la diagnosi e il trattamento delle malattie. In questo contesto, la segmentazione di immagini diagnostiche rappresenta un passo molto frequente che può determinare il successo di queste procedure. Sfruttando strumenti di intelligenza artificiale, nuovi efficaci metodi per eseguire segmentazione automatica sono in fase di sviluppo. Questo lavoro di tesi sfrutta le architetture di apprendimento profondo (Deep Learning) per ottenere una segmentazione 3D veloce, accurata e automatica delle ossa dell'articolazione del ginocchio, in pazienti affetti da artrosi avanzata che subiscono un intervento impianto di protesi totale di ginocchio. Questa operazione chirurgica ha l'obiettivo di alleviare il dolore, migliorare le funzionalità e ripristinare la corretta meccanica del ginocchio, compromessa dal forte grado di usura delle ossa dell'articolazione. Con il crescente tasso di incidenza di tale intervento, sono nate nuove soluzioni, che comprendono una pianificazione preoperatoria personalizzata che richiede la ricostruzione digitale dell'anatomia del ginocchio del paziente. Sulla base delle superfici ricostruite, vengono realizzate maschere di taglio ad hoc, utilizzate per eseguire l'operazione con una minore invasività rispetto agli approcci tradizionali. La segmentazione automatica è ottenuta attraverso modelli Deep Learning che imparano a mappare i dati di input a degli output desiderati, ovviando alla necessità di estrarre manualmente le caratteristiche dei dati di addestramento. Questi modelli utilizzano un ampio set di dati per apprendere la funzione di mappaggio, confrontando ciclicamente l' output temporaneo con gli output di riferimento presenti nel dataset, attraverso una funzione di costo, e aggiornando progressivamente i loro parametri in base all'errore calcolato. Il particolare tipo di architettura sfruttata in questo lavoro sono le reti neurali convoluzionali (CNN). Le CNN permettono di elaborare grandi volumi di dati mantenendo l'invarianza spaziale delle immagini e senza perdere la

connettività locale tra i voxel, grazie all'operazione di convoluzione che viene eseguita attraverso piccoli kernel i quali imparano ad estrarre le diverse caratteristiche rappresentative. Questi approcci automatizzati risultano quasi sempre più accurati dei metodi tradizionali di soglia o degli approcci semi-automatici, che non considerano l'informazione spaziale e contestuale dei dati e spesso falliscono quando i bordi delle strutture non sono ben definiti.
Nel 2015 è stata introdotta un'innovativa architettura convoluzionale per la segmentazione di immagini biomediche, che si è rapidamente affermata come nuovo punto di riferimento. La Unet è una rete convoluzionale che comprende un ramo di codifica e sottocampionamento che estrae progressivamente le caratteristiche dai dati di input e un ramo di decodifica che, mediante strati deconvoluzionali, permette di recuperare l'alta risoluzione spaziale iniziale. Il processo di sovracampionamento sfrutta l'integrazione multi-scala per concatenare l'uscita dello strato di codifica al corrispondente strato deconvolutivo, mediante l'uso di connessioni dirette.
Questo lavoro di tesi si concentra sulla segmentazione delle ossa femore, tibia, rotula e perone. Un dataset di 259 volumi tomografici, forniti in forma anonima da MEDACTA International SA (Castel San Pietro), è stato suddiviso in dataset di addestramento (75%), di validazione (15%) e di test (10%). Alcune pre-elaborazioni sono state eseguite al fine di ritagliare e ricampionare i dati alla dimensione di $192 \times 192 \times 192$, per creare volumi binari di riferimento per ciascuna delle anatomie di interesse. Il lavoro si è poi sviluppato in due fasi. In primo luogo, la Unet è stata addestrata e utilizzata per confrontare 5 funzioni di costo scelte, per capire quale fosse la più efficace. Le funzioni di costo sono le seguenti: Dice Loss, Focal Loss, Exponential Logarithmic Loss, Double Cross Entropy Loss e Distanced Cross Entropy Loss. L'ultima funzione assegna grande importanza ai voxel di contorno delle ossa. Nella seconda fase di questo lavoro, sfruttando i risultati ottenuti nel confronto, è stata sviluppata una nuova architettura con l'obiettivo di migliorare le prestazioni di segmentazione: CEL-Unet. Questo modello mantiene la stessa configurazione di codifica della Unet, e introduce un'innovazione che riguarda la parte di decodifica. CEL-Unet comprende un ramo di decodifica aggiuntivo, chiamato ramo Edge, che produce mappe di segmentazione dei contorni ad alta risoluzione e che procede in parallelo al ramo originale Mask. Le informazioni decodificate nel ramo Edge vengono rifinite dal modulo Pyramidal Edge Extraction (PEE), utile per l'estrazione multi granulare delle caratteristiche dei bordi, e vengono integrate attraverso connessioni verticali dirette al percorso Mask, che genera le mappe di segmentazione finali. La funzione di costo corrispondente include due funzioni, una per ogni output della rete (Mask e Edge), che insieme costituiscono

la cosiddetta Combined Edge Loss (CEL).
L'accuratezza delle segmentazioni è stata valutata con gli indici di Jaccard, Precsion e Recall che consentono di tenere conto degli errori di sovra e sotto-segmentazione. Inoltre, la distanza di Hausdorff e la radice dell'errore quadratico medio (RMSE) sulle distanze superficiali sono stati utilizzati per valutare ulteriormente il grado di corrispondenza tra superfici ricostruite e i riferimenti. Le seguenti 4 regioni localizzate, che rappresentano le aree più critiche, sono state estratte e analizzate singolarmente: condilo destro, condilo sinistro, troclea femorale e piatto tibiale. La CEL-Unet ha superato tutti gli altri modelli basati su Unet, raggiungendo i valori più alti di Jaccard di 0,97 e 0,96 rispettivamente su femore e tibia e minimizzando la distanza di Hausdorff e il RMSE nelle analisi sia globali che locali.
I tempi di addestramento molto elevati (fino a 65 ore) e gli alti requisiti di memoria hanno rappresentato le principali complicanze tecniche, dal momento che l'intero lavoro è stato sviluppato sulla piattaforma gratuita ma limitata di Google Colab. Tuttavia, la segmentazione 3D basata sull'apprendimento profondo si è rivelata estremamente efficace e la nuova, intuitiva architettura CEL-Unet ha fornito risultati molto promettenti per il presente riconoscimento osseo, significativamente complicato dalle gravi condizioni patologiche delle ossa. Secondo i risultati di questo lavoro, questi algoritmi automatizzati potrebbero rivoluzionare la sanità moderna, costituendo la base per strumenti di supporto veloci e intelligenti, atti a ridurre i costi e i tempi di molte procedure e a promuovere un approccio personalizzato alla cura del paziente.

Abstract

Modern medicine is developing in parallel with novel technologies, and new data-driven solutions are gaining interest and effectiveness like never before. Among others, diagnostic images represent a crucial source of information for planning interventions, diagnosing and treating illnesses, with image segmentation being a very frequent step towards success of this procedures. Leveraging artificial intelligence tools, new cheaper and effective ways of performing automatic segmentation are being developed. This work of thesis exploits deep learning architectures to achieve fast, accurate and automatic 3D segmentation of bones in the knee joint, in patients affected by severe osteoarthritis who undergo to PSI-based Total Knee Arthroplasty. This surgical operation entails the implantation of a knee prothesis to relieve pain, improve functionalities and restore knee mechanics of worn out bones. With the increasing rate of incident of such intervention, new personalized solutions were born, with customized pre-operative planning that requires the digital reconstruction of patient's knee anatomy. Based on the reconstructed surfaces, personalized cutting jigs are manufactured and used to perform the surgical operation with a much less invasivness than traditional approaches.

Automatic segmentation is obtained through trained deep learning models that learn how to map input data to some desired output representations, obviating the need of extracting hand-crafted features from the data. These models use large datasets to acquire the ability to perform the task, by cyclically comparing their temporary output with the corresponding reference through an objective function, and by progressively updating their parameters based on the error computed. The particular type of architecture exploited in this work are the Convolutional Neural Networks. CNNs allow to process large volumes of data maintaining the spatial invariance and without losing the local connectivity between voxels, thanks to the convolution operation that is performed throughout small kernels that learn to extract different feautres. Automatic image segmentation achieved with these approaches almost always outperforms the

traditional thresholding or semi-automatic methods, that do not consider spatial and contextual information and frequently fail when structures' boundaries are blurred. In 2015, an innovative convolutional architecture for medical image segmentation was successfully introduced, establishing a new benchmark for this task. The Unet is a feed-foreward convolutional network that comprehends an encoding, down-sampling branch that progressively extracts features from input data and a decoding branch that, by means of deconvolutional layers, allows to recover the initial fine-grained spatial resolution. The upsampling process exploits multi-scale feature fusion to concatenate the output of the encoding layer to the corresponding deconvolutional layer, by the use of skip connections.

This work of thesis focuses on segmentation of femur, tibia, patella and fibula anatomies. A dataset of 259 CT volumes provided in anonymous form by MEDACTA International SA (Castel San Pietro) was split in training (75%), validation (15%) and test (10%) sets. Preprocessing was performed in order to crop and reshape the volumes to the dimension of 192×192×192 and to create reference binary volumes for each of the interested anatomies. After this, the work was developed in two phases. In the first place, the Unet was trained and used to compare 5 chosen loss functions, to understand from which the learning algorithm could benefit the most. The loss functions are the following: Dice Loss, Focal Loss, Exponential Logarithmic Loss, Double Cross Entropy Loss and Distanced Cross Entropy loss, with the last one assigning great importance to boundary voxels. In the second phase of this work, leveraging results of the comparison, a novel encoding-decoding architecture was developed with the aim of enhancing segmentation performances: CEL-Unet. This model maintains the same encoding configuration of the Unet, and introduces an innovation that regards the decoding path. CEL-Unet includes an additional decoding branch, called Edge branch, that produces high resolution boundary segmentation maps and runs in parallel to the original Mask branch. Information decoded in the Edge branch is enhanced by Pyramidal Edge Extraction (PEE) module, for mining multi-granularity edge features, and is integrated through vertical skip connection in the Mask path, that generates the final segmentation maps. The corresponding loss includes two functions, one for each of the two outputs of the network, namely Mask and Edge, yielding the so-called Combined Edge Loss (CEL) function.

The accuracy of the segmentations was assessed with Jaccard, Precsion and Recall metrics that allow to account for over- and under-segmentation errors. Hausdorff distance and Root Mean Squared Error were also used to further evaluate the matching between reconstructed and target surfaces. 4 localized regions that represent the

most critical areas were extracted and analyzed singularly, which are right condyle, left condyle, femur trochlea and tibial plateau. CEL-Unet outperformed all other Unet-based models, reaching the highest Jaccard values of about 0.97 and 0.96 on femur and tibia respectively and minimizing the Hausdorff distance and the RMSE in both global and local analyses.

Very high training timings (up to 65 hours) and memory requirements represented the main technical challenges, since the whole work was developed on the free but limited Google Colab platform. However, deep learning-based 3D segmentation was found to be extremely effective and the novel, intuitive CEL-Unet provided very promising results for this task, significantly complicated by the severe pathological condition of the bones. According to the outcomes of the present work, these automated algorithms could really revolutionize modern healthcare, building fast and intelligent support tools in order to decrease costs and timings and foster a personalized approach to patient care.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Clinical evaluation of medical data is gaining huge interest and importance day after day, starting to be one of the most fundamental driving concepts in modern healthcare systems. The great amount of data that is constantly recorded and stored by hospitals and clinical structures constitutes a continuous source of information that could hide many desired solutions and responses. At the same time, it requires a lot of work to be performed in order to discover such innovative solutions and prevent the waste of all resources used for the organization and the storage of this flow of data.
Medical images constitute a big part of this ensemble. Technologies used to acquire, process and visualize internal structures and anatomies have progressively decreased their costs and timings, allowing the collection of structured high-resolution information that every day helps physicians in the most varied tasks. Making diagnoses, tracking the process of an ongoing illness, planning of surgical interventions and treatments of diseases are some of the clinical procedures for which the support of medical images becomes primary. For this reason, in the last decades, the analysis of biomedical images has developed widely and rapidly, with the focus on some particular tasks, including medical image segmentation, central topic of this work of thesis.
Medical image segmentation is a procedure that has the aim of subdividing the image into its main constituent regions, usually done to highlight the desired anatomies and to discard irrelevant contextual information. It can be applied in a wide range of medical domains ranging from orthopaedics, oncology, neurology even up to histology. Many methods are present to perform such a task, but in the last years, the development of artificial intelligence models proved how new automatic frameworks that leverage this technology can overcome traditional approaches, which are often

more expensive, time consuming and less robust. Relying on many recent studies and researches going on on the topic, this work of thesis presents a new innovative Convolutional Neural Networks (CNN) architecture for segmentation of the knee bone anatomy, after the investigation of a well-known model, the Unet [1]. This network can be considered as a new state of the art in automatic segmentation frameworks. However, in a remarkable recent study [2], authors argued that no deep learning model is ready-to-use for every application, stating that very similar architectures may lead to very varying results across datasets. The need of tackling each problem through the careful tuning of network's hyper-parameters remains strong, achievable with a customization of the training setup and the choice of correct loss functions. These consideration led to the idea of developing a new architecture, tailored for the desired task and with the aim of improving segmentation accuracy provided by traditional methods, semi-automatic methods and also by the classical Unet.

Men have always used machines as "stupid" tools, created to achieve tasks in order to help or go beyond physical human possibilities. The idea of such machines has recently completely changed, which is the main consequence of the origin of the field of artificial intelligence, or machine learning. Today computers can help humans in the process of thinking, assisting experts in difficult decision making problems [3] and providing different points of views of given issues. The main goal of artificial intelligence algorithms, also called machine learning algorithms, is to make predictions, mapping input data to some outputs through a learnt representation, exploiting great computational efficiency of computers and new hardware. Algorithms learn to make predictions because they are trained to do so, by processing the huge amount of data that our era provides. Mostly every kind of structured data features can be learnt by an algorithm, that is why machine learning has lately been applied to a vast number of fields: from economy to logistic, from sports to speech and face recognition, from automotive to medicine.
As in all other contexts, medical issues turn out to be very delicate and hence more difficult to treat compared to some others. With machine learning coming to play, this fact gets even more enhanced by the generation of a direct interaction between intelligent computers and people's health. However, this is also the reason that makes it one of the most fascinating and challenging discipline, not just for physicians, but also for engineers. Algorithms applied to medicine can really help doctors in decision-making, in reducing protocols time and costs and developing new directions of research. The next main issue is to implement those methods in a clinical environment, with

a simple, easy-to-use interface for the physician. If achieved, this could completely revolutionize healthcare. In this work, it is explored the possibility of applying deep learning algorithms with the aim of automatically segmenting the knee bones in CT scans. This is a challenging task that can be applied to facilitate preoperative planning of interventions like Total Knee Arthroplasty, where important preliminary observations and decisions can drive the surgical intervention towards success. Global context and motivations for this work of thesis are explored in the second chapter.

There is currently a huge number of active research topics regarding AI, all with the common aim of automating some type of function. Some examples are understanding and recognizing speech and images, competing in strategic game systems, self-driving cars, interpreting complex data and making diagnoses in medicine. However, until now the expansion of this discipline is mostly due to the success of CNN. The term neural networks refers to the structure of these algorithms that are built to mimic the physiology of the billions of neurons inside the human brain. Among these, different kinds of Artificial Neural Networks (ANN) are found in literature, depending on the type of data that must be treated, the application and the desired output. This thesis work will focus on one particular type, the afore-mentioned CNNs. The term convolutional refers to the convolution operation, that in this application made it possible to treat and process big amounts of data, such as images and video, which would be infeasible to process with the classical Multi Layer Perceptron (MLP) models, that rely on simple affine combinations. Models like CNNs can already outperform humans in different tasks, like object and face recognition and image classification [4] [5]. That is because, in the era of the smartphones, tons of digital images of any kind are generated every moment and are easily accessible. Therefore very big and significant datasets have been created and used to train dependable models that can almost always tell us which animal is inside a picture or where a traffic light is located inside a shot. Indeed, together with the architecture, datasets are what make the difference in the way a model learns to make reliable predictions. Rich, structured, well-labeled and truly representative datasets are the essential starting point for building algorithms able to learn features from seen data and to recognize those attributes in new, unseen and consistent data that need to be classified or somehow predicted. The third chapter of this work is dedicated to deep learning, with a general overview of it, and to the convolutional neural networks, central topic of this study.

There are many possible applications of the convolutional neural networks, one

of which is the so-called semantic image segmentation, which is what this work of thesis is mainly about. Image segmentation is the task subdividing an image into its constituent region objects, to obtain a representation easier to analyze. It consists in creating a mask that can highlight the region of interest on top of the background of the image. It is not just about understanding what is inside a picture, but also about where it is and how it is shaped. It is a classification task, where each pixel in an image is assigned to a certain predefined class, and all pixels grouped together share some important characteristics in terms of intensity, texture or color. Segmentation applied to medical images is being studied deeply in order to obtain algorithms that can automatically recognize anatomical structures of all kinds and for many different purposes with always higher accuracy. Regarding medical images, there is not a universal algorithm used for automatic segmentation. The choice of the method applied depends on the imaging modality, part of the body analyzed and goal of the study. In certain cases, a manual segmentation still represents the gold standard, even if this process is tedious, time-consuming and prone to errors. Fourth chapter will focus on image segmentation, providing a brief overview of the traditional methods to perform segmentation and a detailed description of a famous deep learning model, the Unet. This is followed by the presentation of main developments of that model, with a particular focus on recently developed medical image segmentation algorithms.

As mentioned, this work is about segmentation of bones of the knee joint anatomy. The proposed method aims to study an effective approach to segmentation, where the model, during the optimization, is biased and forced to focus on the most representative features of data, through targeted loss functions and a new architecture. Segmentation frameworks presented here aim at the digital reconstruction of all 4 bones anatomies in the knee joint, that are distal femur, proximal tibia, patella and fibula. An additional class is considered for background and soft tissue together, which are irrelevant for our purposes. The work is divided into two main parts: the first one constitutes a comparison of 5 known loss functions that have been used to train 5 models leveraging the Unet architecture. In this way the most suitable and effective function is found among the ones considered. The second part of the study was dedicated to the development of a new architecture that exploits the same encoding-decoding structure introduced by the Unet, with the aim of improving segmentation accuracy, so to have a more faithful reconstruction of the interested anatomies. In the fifth chapter, all methodologies and approaches, together with frameworks used, are presented in detail.

Lastly, chapter six, seven and eight are dedicated to the presentation and discussion

of the results and to the conclusions and future developments of the present study.

The work of thesis was completely developed in Politecnico di Milano, under the department of Electronic, Information and Bioengineering (DEIB). The collaboration between Politecnico di Milano and Medacta International SA was crucial for the possibility to access a dataset of CT scans of the knee joint, provided in anonymous form and composed by almost four hundred volumes.

Chapter 2

Total Knee Arthroplasty

The focus of this work of thesis is on the knee joint anatomy, for patients that undergo to a surgical operation that requires the implant of a prosthesis to restore correct knee joint motion and mechanics. Such an operation is called Total Knee Arthroplasty (TKA) or Total Knee Replacement (TKR), and both femur and tibia bones are interested. Total knee arthroplasty is one of the most cost-effective and consistently successful surgeries performed in orthopedics. Patient-reported outcomes are shown to improve dramatically with respect to pain relief, functional restoration, and improved quality of life [6]. TKA provides reliable outcomes for patients' suffering from end-stage, tri-compartmental, degenerative Osteoarthritis (OA). The knee is the most commonly affected joint plagued by this progressive condition which is hallmarked by a gradual degeneration and loss of articular cartilage. The most common clinical diagnosis associated with TKA is primary OA, but other potential underlying diagnoses include inflammatory arthritis, fracture (post-traumatic OA and/or deformity), dysplasia, and malignancy. The primary goals of TKA are improved stability, range of motion (ROM), function, and pain relief. Appropriate implant alignment and soft-tissue balancing are important factors in achieving these goals. However, the best method by which to achieve proper implant alignment and soft-tissue balance is controversial [7]. Measured resection and gap balancing are two different surgical techniques that are performed to achieve implant alignment and soft-tissue balance. They differ by the method used in the technique to set femoral component rotation. Both methods include to resect proximal tibia and distal femur in order to substitute the worn original bone tissue of the knee joint with metallic and plastic components of the knee prosthesis. Depending on the chosen approach, bones

are cut in different ways, with the aim of restoring knee mechanics and achieve a successful final implant and limb alignment. Significant values of misalignment indeed can lead to postoperative complications with the consequent need of early revision.

2.1 Personalized Surgical Instrumentation and Preoperative Planning

Personalized Surgical Instrumentation (PSI) is a modern technique in total knee arthroplasty, aiming to facilitate the implant of the prosthesis. The idea behind this technology is to customize the intervention and the preoperative plan for each patient, after a detailed and precise acquisition of the anatomical structure of the joint, through preoperative imaging (2D radiographs, Computed Tomography and magnetic resonance). The customization is achieved through the design and the manufacture of patient-specific cutting guides, that allow the surgeon to resect the bones with higher precision and without the violation of intramedullary canal. The improvement of the mechanical alignment and the optimization of operating room time and logistics are among the main advantages given by good clinical results of the Patient Matched Technology (PMT). The goodness of the outcomes has been investigated in [8], 2 years after the intervention, in comparison with results of conventional instrumented total knee arthroplasty: no significant difference between clinical conditions was found. Moreover, a lower rate of post-TKA mechanical alignment over 3 degrees, among patients operated with PMT, was found in [9], in comparison to patients operated with conventional methods. In this clinical scenario, image/volume segmentation plays a fundamental role as it allows the digital reconstruction knee bone surfaces, which will drive the whole procedure of planning and surgery.

An appropriate musculo-skeletal radiological study is essential for planning the procedure to obtain desired clinical outcome. Medical images like CT scans, radiographs and MR images constitute the starting point for surgical planning in knee arhtroplasty performed through PSI and customized implants [10]. Preoperative planning is part of the procedure to simplify the surgery and the positioning of the components, which is very important for achieving the accurate alignment. Most of the knee replacement systems provide a large number of options for treating conditions that are encountered during surgery. The different systems also serve the surgeons variable sizes of implants

in a wide range, and the correct must be chosen according to patient's anatomy and clinical conditions. The main crucial steps in preoperative planning for TKR can be summarized as follows:

- Assessment of knee joint anatomy through image segmentation
- Identification of clinical landmarks on bone surfaces
- Establishing of tibial and femoral resections
- Implant size selection
- Definition of the implant location to restore knee mechanics
- Manufacturing of patient-specific cutting jigs

Segmentation of bones influences the reconstruction accuracy and in turns strongly affects the results of the whole pipeline. It is important to consider that bone surface reconstruction is just the first step of a long and complicated process that includes the rest of the planning, manufacturing and the surgical intervention itself. Whatever error is made at the beginning of the chain is going to be kept along the whole process, influencing any future step. This can cause errors and uncertainties to be accumulated which can degrade the quality of the results sensibly and affect success of the whole work. For this reason, all precautions should be taken and all new promising methodologies should be considered as an attempt to improve results. Accurate, sub-millimetric matching between digital obtained surface and real bone geometry is required to ensure successful outcomes, and the ideal should be to achieve it in an automated fashion.

2.1.1 MyKnee - MEDACTA International

MyKnee is a personalized surgical instrumentation solution patented by Medacta International SA that facilitates and improves the effectiveness of surgeons' preoperative planning for knee arthroplasty. The cutting blocks are designed to fit patients' anatomy and restore mechanical axis at best and are positioned on the bone referring to some distinct references, extrapolated by the geometry reconstruction. The success of the surgery depends on the quality of the matching between the patient-specific resection

jigs, manufactured exploiting the patient bony surfaces attained by segmentation, and true patient surfaces.

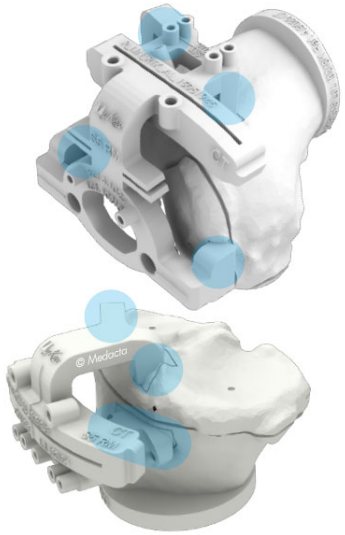


Figure 2.1. MyKnee solution by MEDACTA International. The two cutting jigs hook at the femur and the tibia, facilitating the cutting process.

No intramedullary canal violation is ensured with the use of this technology. This is demonstrated to reduce more than 20% the blood loss during the intervention, with respect to normal jigs [11]. Also, reduction of surgical steps, time and cost are among the great advantages of this technology. As a consequence, an optimization of the O.R. use is found, thanks to the reduction of the surgical time, which can potentially add one extra case per surgery session. [12]

2.2 Work motivation

Millions of TKR surgeries are performed on earth per year and this number is increasing day by day, with the novel designed technologies [13]. TKR surgery is the gold standard method in the treatment of end stage knee arthritis with a high success. The increasing trend of global median age brings to a greater diffusion of this kind of illnesses, strictly related to patients' age. In [14] and [15], authors show the increasing incident rate of TKA in the U.S. year after year and projections for the future have been made and presented. Figure 2.2 shows an example of recorded data and projections for both TKR and Total Hip Arthroplasty. With such a great

increment on the horizon, fast renovation of present technologies is necessary to face and win new upcoming challenges. In this scenario, the process of digitalization of

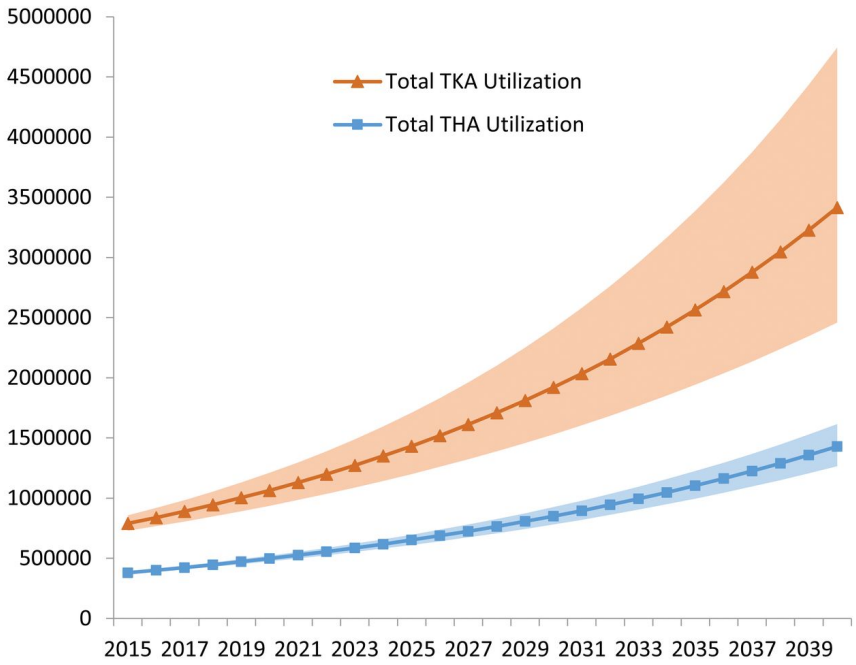


Figure 2.2. Graph from [14]. The projected annual use of primary total hip arthroplasty (THA) and primary total knee arthroplasty (TKA) procedures in the United States from 2015 to 2040. The X-axis shows years and the Y-axis shows the number of annual procedures for primary THA (blue) or primary TKA (orange).

medical images started with the new century resulted in the collection of a huge amount of data that has to be stored and needs to be analyzed. As a consequence, engineers, researchers and physicians can leverage this data to create innovative tools with the aim of reducing costs of healthcare, facilitate care access and reducing risks at the most.

2.2.1 New Solutions

Bone segmentation can be achieved in many ways: automatic software are present on the market, together with some semi-automatic tools. Also, manual work of expert orthopaedic radiologists sometimes remains the gold standard for some applications. However, these workflows usually require a great effort in terms of cost and/or time: this renders very impractical to analyze large cohorts of data following this procedures,

that can be error-prone and suffer from intra-operator variability [16]. The manual delineation work also depends on the experience of the operator. Segmentation is then complicated by non-physiological conditions of patients that undergo such surgery. Pathological disorders affect bone structures and alter the morphology of cartilage and bony surfaces, usually narrowing the intra articular space. Moreover, the frequent formation of osteophytes generates uneven and ragged geometries at the edges, which are difficult to distinguish and to delineate. This makes the segmentation even more difficult, requiring additional checks and validations by other experts.

Given those considerations, the context suggests exploring alternative and cheaper ways to solve the task, leveraging new rising technologies. Some years ago, deep learning was found to be very effective in performing automatic segmentation on digital images, with the ability of discerning different structures and classify them automatically by extracting features like texture, color, intensity, shape and others. Deep neural networks present a large number of layers that learn the representation from the input to output. This achievement is possible thanks to the great development of computational power in the last decade, given by hardware re-discovered very effective for this type of application. In many cases, the performances of CNNs on a wide range of tasks are comparable or even greater than older and classical approaches. In this sense, deep learning is shaping the future also in the medical field because it can help many procedures to become automatized. This is the case of medical image segmentation that is being extensively researched and studied by groups all over the world thanks to its great potential in helping with planning interventions, diagnosing illnesses and also treating patients.
The challenge is to explore all the possible architectures, combining many different elements, and to find the configuration that performs the best on a particular type of data. The goal is to find a model that can really learn features that are relevant for the desired task, and then generalize the behavior acquired on the training dataset onto a set of new data. Intelligent models, able to generalize. So, the structure of the model must be really targeted to the kind of data under inspection. Many complications are on the way and many problems are yet to solve, including the challenge to reach very high levels of accuracy, always strictly required in treating delicate medical scenarios. However, results are already very promising and give a glimpse of the positive impact that these technologies can have on the world of the future, with the intelligent and constant support that deep learning-based applications can provide to humanity.

Chapter 3

Convolutional Neural Networks

3.1 Deep Learning

In image analysis, all methods share the common idea of extracting features from the inputs and use them to produce a classification output. The next step in this procedure, is to achieve this result in an automatic fashion. Great, innovative and promising solutions to the problem are provided by the recent and broad expansion of artificial intelligent applications. These algorithms are revolutionary because they learn from experience, tackling problems based on what they have already seen before. They acquire knowledge of the world building up a hierarchy of concepts, where each single concept is interpreted as an ensemble of simpler ones. Therefore abstract and complex representations are learned from easier factors. In this way, the machines automatically build hierarchical statistical models, without anyone programming exactly what they need to learn. The visualization of these hierarchical set of concepts can be thought as a structure with many layers, that is why this approaches are referred to as deep learning. Deep learning differs from the broader filed of machine learning basically in the representation of data that can be learnt by the models. With machine learning, the data needs to be represented with some hand-crafted features so that models can learn the mapping from these representations to the output. Deep learning models instead are built to directly learn not just the mapping, but also the representation itself, obviating the need of hand-crafted features and often giving better results, generalizing also to new tasks with very little human effort needed.

In general, deep learning algorithms can be divided into supervised and unsuper-

vised: the former regard learning features from labeled data, whereas the latter aim at a pattern analysis. From now on, this work will focus on supervised learning.

3.1.1 Supervised Learning

In a supervised learning task, the dataset consists of a set of training examples, each one containing the input object and the desired output, usually referred to as label or ground-truth. A learning algorithm is chosen to let the model recognise and extract the representative features of the input objects during training. Learning is usually performed by comparing the current output produced by the model with the ground truth provided in the dataset and trying then to minimize the error computed between the two. By performing this procedure repeatedly, the model can progressively update its parameters to enhance each time its ability to perform the desired task. A few guidelines must be considered when approaching a problem with supervised learning. First of all, all training examples contained in the dataset should be representative of the real-world distribution of the samples and should be meaningful for the learning task. Since a finite dataset cannot include all possible data, it should be at least able to provide a realistic representation of the samples distribution, to ensure a correct learning process.
In the second place, the chosen model should be tuned and well calibrated for the given task. Too big models tend to overfit the data they see, meaning they don't learn representative concepts and features to classify the data, but they learn the actual training data. This decreases the ability to translate the performance on new, unseen data. On the contrary, too small models can't provide enough capacity to learn and store all the necessary information, which brings to unsatisfactory results.
Lastly, a few examples should be kept out of the training data and used as a test set, to evaluate how the trained model can perform the task on new data.
Supervised learning methods deal mainly with two categories of problems: regression and classification. Regression is the process of predicting a variable with continuous values, whereas a classification task predicts only discrete values or categories into which the data is separated. Many different algorithms have been developed for both kinds of problems, from support vector machines, to classification trees, to neural networks. As mentioned, this work will focus on a special type of neural networks, the convolutional neural networks, which will be presented in the following paragraphs.

3.2 Basics of Convolutional Networks

In the last few decades, neural networks have been discovered extremely useful and powerful in perform classification and segmentation tasks. The neural networks are models that stack layers one after another to form a sequence going from the input layer to the output one, through a set of so-called hidden-layers. Each layer is made of single entities called neurons. These neurons perform linear or non-linear transformations on input data and pass the information to subsequent layers, until the last one is reached, where the output of the network is produced.
In the case of image and video analysis, particular kinds of neural networks are used, called Convolutional Neural Networks. In fact, whereas Fully Connected Networks perform well in raw numerical data analysis, they completely fail when dealing with images. Some fundamental differences must be considered between numerical datasets and image datasets. First, images and volumes are represented by 2D and 3D vectors, composed by single pixels or voxels, each of which can take up to 3 values in the case of colored images. This fact enormously increases the size of input data, making infeasible the idea of storing a set of parameters dedicated to each single unity. Too many connections would be required and the computation would be too expensive. Secondly, spatial information is crucial in this case and has to be considered somehow. Treating each pixel (or voxel) as a single isolated input entity would destroy any global contextual information. Moreover, each element is strictly linked with its surroundings, and a good analysis must account for that.
Convolutional neural networks have the same global structure of MLP models, with sequences of layers passing and transforming information from the input to the output, learning the parameters that define the model through backpropagation algorithm. The difference stays in the operation that is performed inside each layer of the network, which is, in this case, the convolution. This mathematical tool allows to sensibly reduce the amount of parameters inside the network and fatally increase the efficiency of computations. Together with that, it allows to process the images as a whole, learning the spatial relationships between single elements in the input image. This operation is widely applied in many fields of engineering and it is the basics of computer vision algorithms.

3.2.1 Discrete Convolution

Discrete convolution is an operation performed between two discrete entities, an input and a filter, usually called kernel. The kernel must have the same number of dimensions of the input. For simplicity, we will refer to 2 dimensional objects, but this concept is easily extended to any dimension. The idea is to use a filter K and slide it over input image I, so that it covers all the possible positions. In each location, a multiplication is done between all the elements of the kernel and all the overlapping values of the input image, and everything is summed up. This sum of products will be the value of the output image, in the central position of the current kernel location. The mathematical expression can be written in this way:

$$(I * K)_{u,v} = \sum_{i=-h_1}^{h_1} \sum_{j=-h_2}^{h_2} I(u+i, v+j) * K(i,j) \tag{3.1}$$

where kernel K has size $[2h_1 + 1; 2h_2 + 1]$ and the following configuration:

$$K = \begin{pmatrix} k_{-h_1,-h_2} & \cdots & k_{-h_1,h_2} \\ \vdots & \ddots & \vdots \\ k_{h_1,-h_2} & \cdots & k_{h_1,h_2} \end{pmatrix} \tag{3.2}$$

Instead of learning weights of the affine linear transformation as it happens in FCNs, here, the values of the filters are learnt and optimized in order to extract the most significant features from input data. Many filters are learnt in each convolutional layer, each of which produces one output. It is important to notice that the convolution operation characterizes the network with some very important aspects:

- **Weights sharing**: much fewer parameters have to be defined with respect to FCN. By sliding the kernels on the image, all the pixels share the same weights, defined by the values of the filters. Hence, the amount of parameters just depends on the number and the size of the filters. This sensibly decreases the computational complexity and makes the operation cheaper and faster.
- **Spatial invariance**: spatial consistency is maintained inside the network and the structures are processed independently of the location they have in the

image. Any target object is recognized as such regardless its location in the image.

- **Local connectivity**: CNNs take advantage of local spatial coherent properties of images, and enforce spatial connectivity patterns between neurons in adjacent layers. The value of each neuron in one layer only depends on few neurons in the preceding layer. The network is indeed not fully connected, but, conversely, neurons are only connected to those neurons in the next layer that are spatially close.

The convolutional layers perform convolution of inputs with certain number of filters, producing number of outputs equal to the amount of convolved filters. The outputs are referred to as feature maps. In figure 3.1 an example is presented, showing the original input slice and the 8 feature maps of the first (above) and the last (below) convolutional layers of a Unet segmentation model. Parameters of each filter are learnt to extract different pieces of the information encoded in the input image. First layers focus more in the extraction of low-level features, like edges and blobs, while deeper layers are able to learn abstract and higher-level features, like colors and shapes and representative patterns, bringing to the definition of the output segmentation masks.

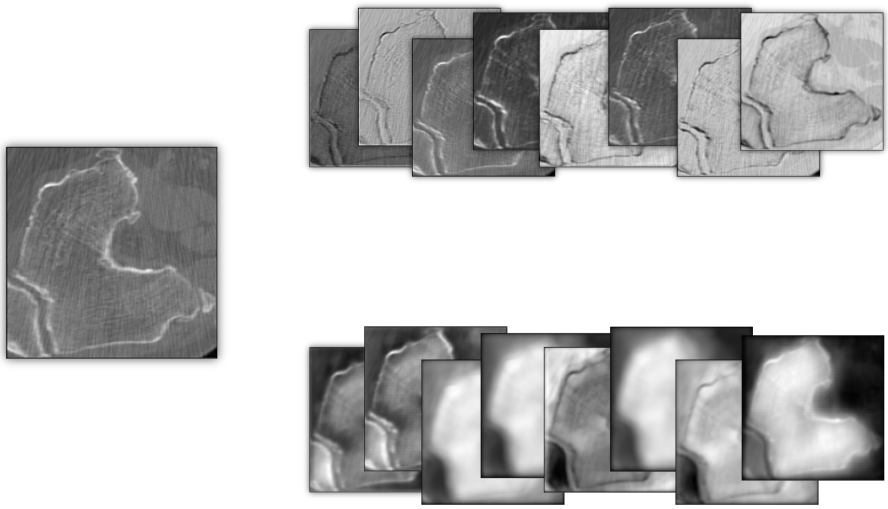


Figure 3.1. Example of 8 feature maps from the first and the last convolutional layers inside a Unet architecture.

3.2.2 Pooling Layers

The configuration of CNNs is defined with the alternation of convolutional layers and pooling layers, to progressively reduce the size of the input. Pooling layers act like down samplers, producing representations at a lower resolution than the input image or volume, maintaining meaningful features and discarding what is less relevant. Pooling operation is performed level after level on the output feature maps of the convolutional layers. These feature maps present the limitation to record the precise location of features in the input. Small changes in the position of such features result in different feature maps in output, which harms the ability of the network to recognize targets regardless their location. With down sampling instead, these small changes result in the same pooled feature maps, due to the coarser representation of low resolution tensors. Stacking pooling layers after convolutional layers allows to obtain a system where the progressive loss in spatial resolution enforces the model in the ability of being invariant to local translations. Also, computational costs are decreased due to the reduced dimension of data. Moreover, the gradual reduction of image size helps filters in deeper layers to focus on larger receptive fields and capture the global context. The information contained in the input image is maintained and passed along the network but, step after step, it is encoded in a gradually reduced space. For instance, if the input size of the network is 200×200, the same information gets encoded in deep layers in a down-sampled size that could be 20×20. In deeper layers then, the same filter size acts on patches of the feature maps that represent much larger space of the real input data. Pooling can be performed with two different operations: Max-Pooling and Average-Pooling. Kernels of specific dimension, typically 2×2 kernels with strides of 2 are used. Examples of Max and Average Pooling are show in figure 3.2.

There is another important aspect to be taken into consideration. Simple, low-level features are extracted by kernels in the first convolutional layers. When the network goes deeper, the representation of such features becomes richer and richer, as the simple attributes are combined into more complex information that need to be captured by the network and in turned encoded and passed to next layers. For this reason, the number of filters inside the network is increased for each time the net goes one step deeper. Usually, each time the data is halved in size with a pooling operation, the number of filters is doubled, which brings to the production of double the number of feature maps with respect to the previous layer. Following this approach, not just

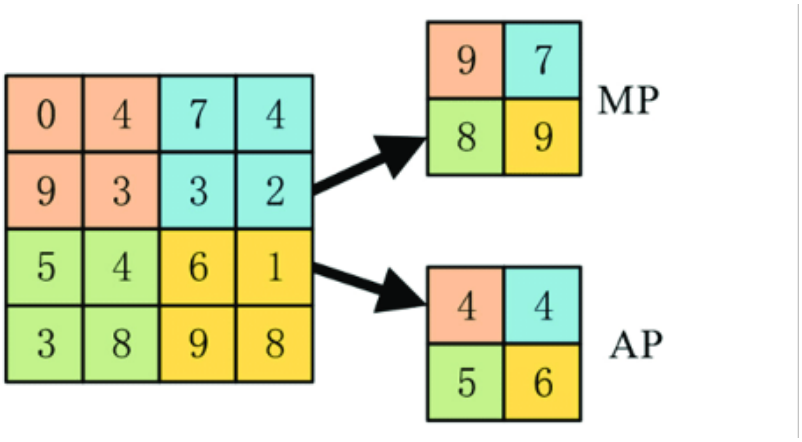


Figure 3.2. Example of 2×2 Max pooling and Average pooling operations on 4×4 input matrix.

simple features are encoded and learnt by the model, but also abstract high-level features in deeper layers. This fact helps the network to increase its generalization power.

3.2.3 Activation Functions

Another important component of CNN architectures are activation functions. They are applied to the outputs of convolutional layers, mapping values depending on the morphology of the function and basically deciding whether a neuron will fire or not. These functions serve for introducing non-linearities to neurons, in order to better encode the complex representation from the input to the output of the model and recognize and learn patterns in data. Also, the generalization ability of the networks benefits from this. Some different types of activations can be used, depending on the task and the architecture choice. Bounded activations squeeze input values into an output range: $[0; 1)$ for Sigmoid, $(-1; 1)$ for hyperbolic tangent, for example. On the other hand, unbounded functions can map inputs up to infinite.
Although some smooth non linear functions were mathematically inspired by the biological neuron behavior, most of current methods are using the ReLU function. Authors in [17] and [18] argue that it achieves better results in deep networks as it makes the activation sparse and more efficient. This activation maps negative inputs to zero and positive inputs with an identity transformation, bringing to the following mathematical formulation, and graphic representation:

$$y = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

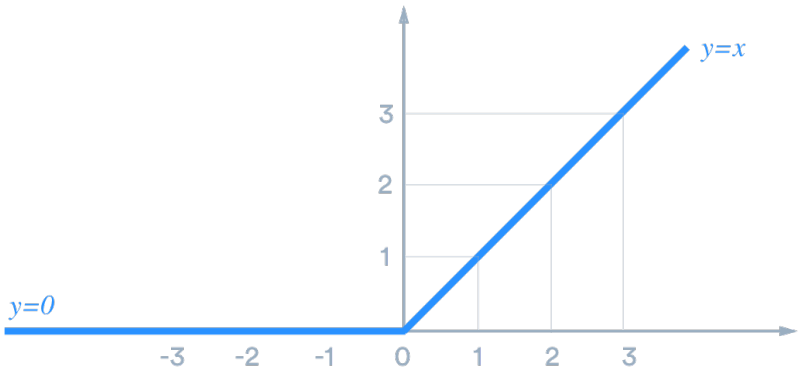


Figure 3.3. Graphical representation of ReLU activation function.

In [19], authors claim that ReLU activation function can improve the learning speed of deep neural networks. Since gradient of ReLU function is constant at positive values, this fact solves the problem of vanishing gradients that can occur when number of layers starts to be high.

3.2.4 Softmax Layer

Softmax layer is used in classification problems to produce class scores at the output of the network. Since segmentation problem is nothing else than a pixel-level classification task, softmax usually represents the last layer of segmentation CNNs. It produces probability mask where values specify the likelihood of each pixel to belong to a single class:

$$Pr(Y = c) = \frac{e^{X_c}}{\sum_{c=1}^{C} e^{X_c}} \tag{3.3}$$

where X_c is the value of a pixel in the input image, C is the number of classes and Y represents the pixel in the output image.

3.2.5 Loss Functions

The loss function in neural networks computes the prediction error, comparing the output of the model and the ground truth labels of the train or validation dataset. At each step, e.g. at the end of each minibatch, the loss is computed together with its gradient with respect to the network hyperparameters. This serves to the optimizer which updates filters and biases of the model, with the aim to minimize the loss function and improve the prediction. Suitable loss functions should be differentiable, to allow computation of the gradients, and must reach global minimum when the prediction matches the ground truth. Different loss functions can be used, also depending on the nature of the output of the network.

- CROSS ENTROPY LOSS
  With probability outputs, e.g. softmax layer as the last one, an appropriate function is the cross entropy loss, which computes the cross entropy between ground truth segmentation masks in $\{0, 1\}$ and probability scores in [0,1]. Cross Entropy (CE) is computed for all pixels and all classes and every value is summed up to obtain a scalar that indicates how far model is from making the right prediction. Formulation of such a loss between $\hat{y}$ probability mask and y ground truth mask is as follows:

$$CE(\hat{y}, y) = -\frac{1}{N}\sum^{N}(\sum_{c=1}^{C} y_c \cdot \log \hat{y_c}) \tag{3.4}$$

  where N is the number of pixels in the image, C is the number of classes, y_c is a value in $\{0, 1\}$ to indicate whether the pixel belongs to class c or not and $\hat{y_c}$ is the probability value assigned to the same pixel to belong to class c.

- DICE SIMILARITY COEFFICIENT
  Another frequently used loss function for segmentation tasks is the Dice Similarity Coefficient (DSC), which evaluates the degree of overlapping between ground truth mask and predicted mask. Considering figure 3.4, that shows the superimposition between the two masks, we can consider True Positives (TP) and True Negatives (TN) as the correctly classified pixels, while False Negatives (FN) and False Positives (FP) as misclassified units. The DSC is then

computed as follows:

$$DSC(\hat{y}, y) = \frac{2 * TP}{2 * TP + FN + FP} \tag{3.5}$$

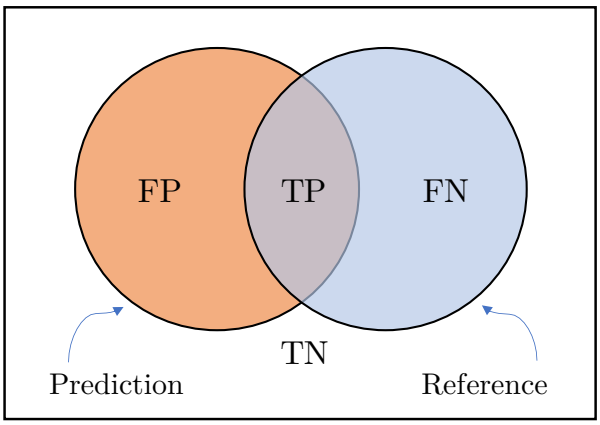


Figure 3.4. Graphical meaning of dice similarity coefficient.

Chapter 4

Segmentation of medical images

Image segmentation is the task of recognizing different structures inside an image and delineate their boundaries, in order to differentiate them from one another and from the background. Such regions can be whatever object, animal, landscape depending on the target of the work. In this thesis, the focus is on medical images, and in particular, on the knee joint anatomy. Segmentation is a classification task where instead of classifying whole pictures or instances, models focus on classifying pixels in an image, or voxels in a volume. A label is assigned to every single voxel in what is commonly referred to as semantic segmentation, and the result is an image where voxels with the same label share certain characteristics. Therefore, image segmentation provides a more meaningful representation of the data and it is a crucial step for fully understanding the content of medical images and for doing diagnosis. After segmentation, all the disjoint regions should be homogeneous with respect to some characteristics and show spatial compactness. Good image segmentation should meet some fundamental requirements:

- Every pixel in an image must belong to a class
- Each region is homogeneous with respect to some characteristics
- No region overlaps are present

Segmentation still remains one of the most studied topic in the literature researches applied to medical images. Different segmentation methods are applicable to face various problems and the most suitable one must be chosen for the specific purpose.

Some of the traditional successful approaches are presented in the next paragraphs. Then the Unet is introduced, together with some innovations applied to it, specifically tailored towards medical image segmentation.

4.1 Traditional approaches

4.1.1 Thresholding

The most straightforward method to perform segmentation is by setting a threshold on the pixel intensity value and classify pixels according to that threshold. It is usually done on grey scale images, where each pixel is encoded in just one value. That makes this approach very easy and fast. The simple operation is shown in the following:

$$S(x) = \begin{cases} 1 & \text{if } I(x) > \text{thr} \\ 0 & \text{if } I(x) < \text{thr} \end{cases}$$

where *S(x)* represents the output segmentation mask and I the input image, with each of its pixels x. This method completely ignores the spatial information given by the location and distribution of the pixels, resulting uneffective on images that present blurred boundaries. The threshold can be assessed by looking at the histogram of the image, inspecting peaks and valleys of the distribution and selecting a threshold value that best clusters pixels into the two desired classes.
However, histograms very often present noisy profiles, which makes it more difficult to create well defined classes to which assign pixels. This usually results in rough segmentation outputs which do not correspond to faithful instance delineation.

4.1.2 Otsu's Thresholding

Otsu's method [20] is the most common one that automatically computes a threshold for image segmentation. The value is assessed by minimization of the variance of the classes and it is applicable only for binary segmentation, meaning when the problem has only two classes. In [21], [22] and [23], authors present some segmentation frameworks

based on Otsu's method applied to medical images and volumes. Being this approach very simple, fast and computationally inexpensive, it can be very practical and useful for binary image segmentation, but also for image inspection and quick, rough target structure delineation. However, simple thresholding for segmentation does not usually provide values of robustness and accuracy required by medical protocols. In addition, its 'blindness', meaning the inability to recognize spatial and contextual information, can make it inappropriate and obsolete with respect to newer, advanced segmentation frameworks. Figure 4.1 shows an example of how Otsu's thresholding method segments a CT slice of the femur.

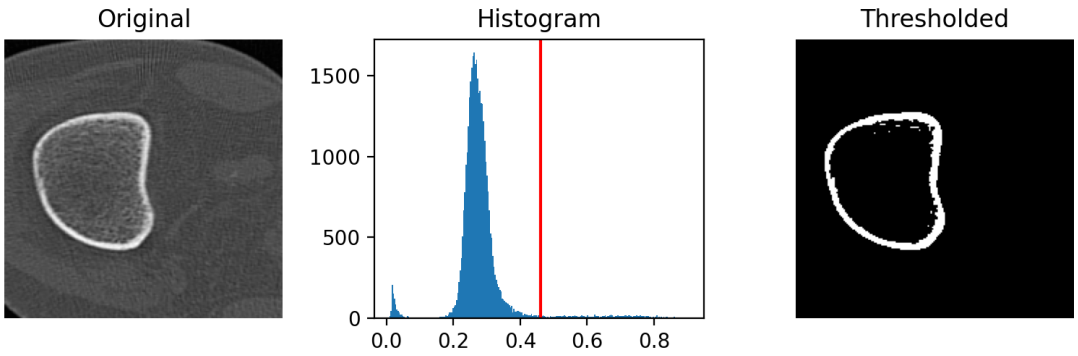
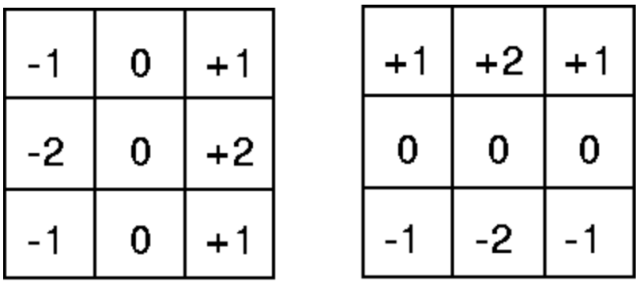


Figure 4.1. The histogram of the original CT slice is used by Otsu's method in order to compute the threshold for the segmentation. The output binary image is shown on the right.

4.1.3 Edge detection

Edge detection is the process that exploits automatic algorithms to locate edges in an image, which can roughly differentiate the structures on the foreground and give a snapshot of the separation between them. Boundaries represent very meaningful features and contain significant information, represented by abrupt changes in intensity values. First and second order derivatives like gradient or Laplacian are used to extract edges in an image. These operations are implemented with discrete kernels that are convoluted with the input image and are able to extract various information about boundaries (horizontal-vertical edges, thin-wide edges).

An advantage of edge detection consists in the reduction of the image size, which makes subsequent processing easier and faster. In the context of image segmentation, it can really discard the meaningless information from the image, maintaining the focus



(a) Sobel operator along x (b) Sobel operator along y

Figure 4.2

on the most relevant part of it. Boundaries are detected from some discontinuities in grey levels, colors, textures, brightness, saturation and other visual features. It is very often part of the processing needed to achieve final segmentation, but it cannot really be considered as a standalone segmentation method: supplementary processing must follow to concatenate the edges into edge chains that better correspond to object boundaries. Edge chains must be then filled in order to create the masks for ultimate segmentation.

4.1.4 Semi-Automatic Segmentation

Semi-automatic segmentation is also known as interactive segmentation, due to the fact that interaction of the user is required in order to generate complete and satisfactory segmentation mask. It is typically composed by three steps:

- User input: a user provides an information, which helps the computer with the computation of segmentation
- Computation: a computer tries to delimit the objects based on the information provided in step 1
- Display output: a computer displays an intermediate segmentation that was computed in step 2

These steps are iteratively repeated and the input can be edited until the user is satisfied with the segmentation result. Depending on user inputs and type of computation, some different methods can be listed, such as Graph-cut [24], Edge-based [25], [26],

Random-walks [27] and Region-based [28] methods, with the last ones comprehending the so called Region Growing methods.
In this case, the initialization is provided with some initial seeds that define the starting point of the segmentation, each one associated with one of the final regions to segment. These may be single pixels or a group of pixels (clusters). Once the seeds are defined, the subdivision is done in steps. For every initial cluster, a comparison is made with adjacent pixels, which can join that specific cluster if they present similar characteristics, in terms of brightness, color, texture, intensity. The computation progresses following various algorithms. In this way, each region accumulates pixels and grows towards its final shape. Some issues accompany this semi-automatic

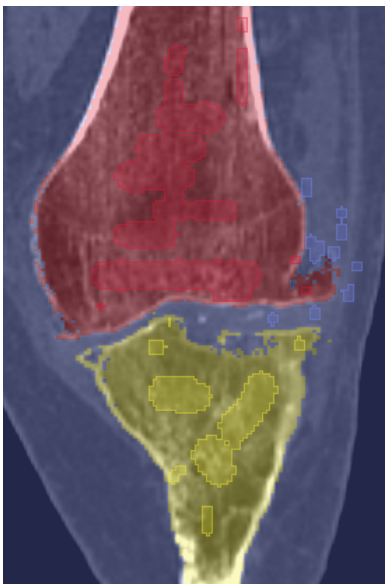


Figure 4.3. The image shows how region growing method segments a frontal CT slice showing femur and tibia bones. Segmentation of the femur is shown in red, while the tibia and the soft tissue are marked respectively in yellow and purple. Seeds defined in the initialization are visible for each class.

segmentation, as the final result strongly depends on the initialization that is given. The more the initial seeds are significant for the specified classes, the more the algorithm will succeed in finding the correct separation. To help the process in growing the regions correctly, many sparse and disjoint seed points may be drawn for a single class, in locations where the algorithm could be more likely to fail. However, this also increases the effort for the initialization, bringing this method further from being automatic and yielding a more manual processing instead.

4.2 UNET

In this section a powerful architecture for image segmentation is presented. Since its publication in 2015, it has become the most popular, used and cited deep learning model for such a task.
Intuitively, we can say that subsampling operation is a good way to provide a model with awareness of what is present in an image. But at the same time, the model progressively loses the information of where it is present, due to the loss of the original spatial representation. This faces the image recognition task, that is why CNNs have been used for some years mainly for image classification, alternating convolutional and pooling layers one after another. This procedure reduces the size of the inputs and eventually produces a vector of probabilities to assign the most likely class to each image, e.g. recognize what it is present in an image, regardless where it is. Image segmentation instead is the process of understanding not just what is in an image, but also where it is. For this reason, the network needs to produce a fine grained segmentation map, where each pixel is assigned to one of the possible classes. In fewer words, the spatial dimension and resolution of the output must be the same as the input. As a consequence, the architecture of the model must adapt in order to recover the original size of the image after the feature extraction, well performed by the alternation of convolutional and pooling layers.
In 2015 Ronneberger et al. [1] developed a model called UNet, with the aim of performing automatic segmentation of medical images using a CNN, which produces a high resolution segmentation mask as output. It is composed by two distinct paths, descending and ascending, also called encoding and decoding paths respectively. The graph of the network presented in the original paper is shown in figure 4.4, where it is clearly visible why the model was named after the letter "U". This model is an end-to-end Fully Convolutional Network (FCN), where just convolutional and pooling layers are present, without any dense layer. The encoding path follows the typical architecture of CNNs. In this case, at each step, two 3×3 convolutions are performed one after the other, each followed by ReLU activation function. Then, a 2×2 max pooling operation with strides of 2 is added to halve the size of the images. At each down-sampling step, the number of filters used to perform the convolutions is doubled, to enlarge the space of features to be extracted from the data. The up-sampling path performs the opposite operations: transposed convolutions are applied to enlarge the spatial representation of data and gradually increase image size, in order to recover the original dimension of the input. Feature maps coming from deep layers represent

abstract attributes and high level information. These are up sampled step after step, and concatenated with feature maps generated at the same level in the down sampling branch. After concatenation, two 3×3 convolutions are performed to unify the information. The residual connections from encoding to decoding path allow to merge global contextual information, coming from deeper layers, with the spatial high resolution information, present in the decoding path at each level. This enables the network to produce fine grained segmentation maps at the output.

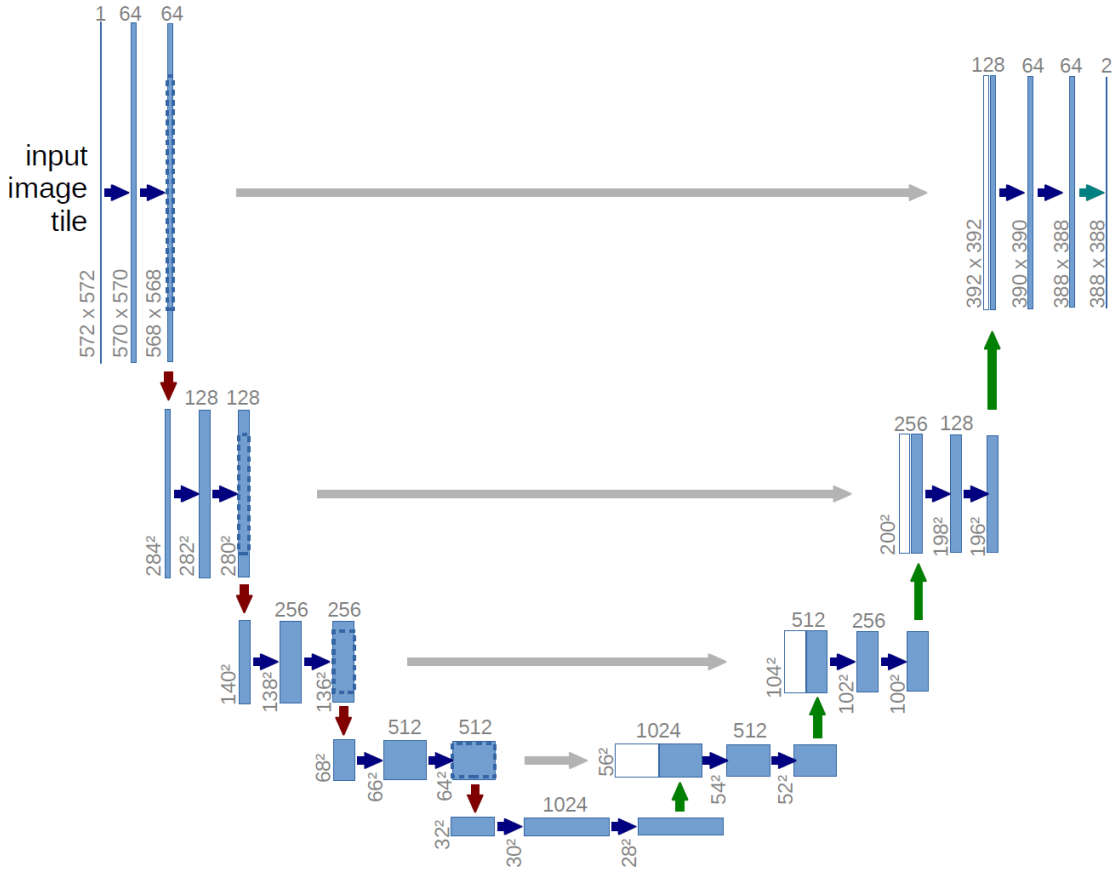


Figure 4.4. Diagram of Unet architecture as presented in the original paper. Blue horizontal arrows stand for 3×3 convolutions + ReLU activations. Downward and upward arrows represent respectively 2×2 max pooling and 2×2 transposed convolutions. Grey long horizontal arrows represent skip connections to integrate feature maps from the encoding branch to the decoding branch.

4.3 Medical Image Segmentation

UNet architecture is the most popular deep learning model used to produce segmentation maps from medical images data. Many studies, in the last few years, investigated its effectiveness in various applications regarding different anatomical compartments, validating its ability to provide accurate results and to sensibly overcome both traditional and semi-automatic methods. For example, both 2D and 3D UNet models have been investigated for segmentation of hand bones in X-ray images [29], mandibular bones in cranio-facial CT [30], femur in CT scans [31] and major skeletal bones in whole-body CT scans [32]. All such papers focused mainly on binary segmentation of bones against the image background, used low resolution data, disregarded the effects of large bone deformations and osteophytes on the segmentation quality. Some modified versions of the UNet are also found in the literature, each one trying to facilitate the process of learning the mapping relationship between pixels in the feature maps and to improve recognition capabilities of target structures.

4.3.1 Attention Modules

Attention mechanism was initially designed in the context of Neural Machine Translation and it is now used in various problems like image captioning, classification and segmentation. It is a method that facilitates the flow of useful information inside the network and discards the less relevant part of it, bringing to a more targeted representation.
In [33], for example, authors propose an end-to-end convolutional neural network called Channel-UNet, which takes UNet as the main structure of the network and adds spatial channel-wise convolution in each up-sampling and down- sampling module, to more clearly distinguishing the tumors from the liver tissue. Spatial channel-wise convolution can be thought as a sort of attention method, where the spatial channel-wise kernel slides across the direction of the feature maps, to better learn the representation of each feature map inside each layer and to focus on the most significant ones. Another interesting attention approach is found in [34], where both spatial and channel attention modules are added to the model in order to emphasize meaningful features along those two principal dimensions. In this way, each one of the branches can learn ‘what’ and ‘where’ to attend in the channel and spatial axes respectively. As a result, this modules efficiently helps the information flow within

the network, by learning what to emphasize and what to suppress. In [35], authors leverage this attention module to perform pancreatic segmentation with great results.

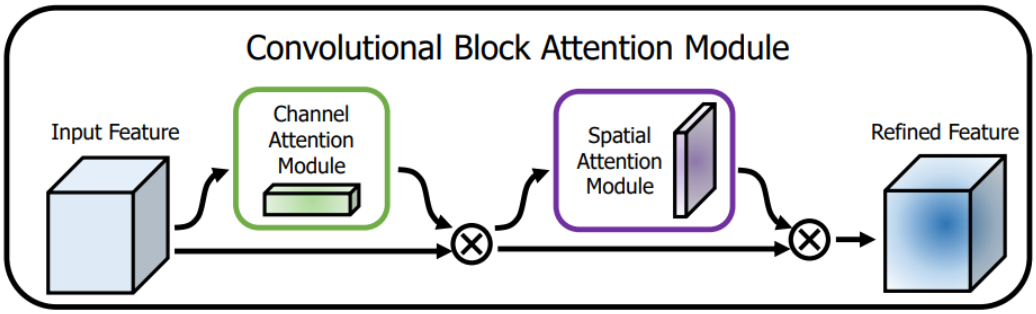


Figure 4.5. Overview of CBAM (Convolutional Block Attention Module). The module has two sequential sub-modules: channel and spatial.

4.3.2 Residual Connections

Training deep neural network is more difficult and challenging than training simpler and shallower models. With the great increase of the number of layers, the gradients inside the network more easily decrease their values during back-propagation, and sometimes they completely vanish, due to the repeated multiplications that may make the gradient infinitively small. Hence, as the network goes deeper, its performance gets saturated or even starts degrading rapidly. This problem was faced in 2015 by He et al. in [36], that introduced the ResNet architecture. The core idea of ResNet is to introduce a so-called "identity shortcut connection" that skips one or more layers, as shown in figure 4.6. This structure helps the network to maintain the gradients during propagation of information and to learn residuals and identity mapping, which turns out to be easier than to learn the actual representations.

This idea can be also transposed to segmentation networks, by introducing skip connections between convolutional layers to build the so-called residual blocks. This is usually done in the segmentation networks to facilitate propagation of useful information and very often contributes to enhance the segmentation performances [2]. Different interpretations and implementations can be found, one of which is the DenseNet [37]. Here, the ResNet idea is taken to the extreme to build the so-called Dense Blocks, where each layer is connected to every other layer in a feed-forward fashion. For each layer, the feature-maps of all preceding layers are used as inputs,

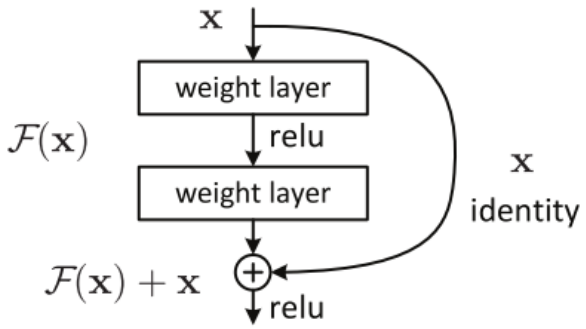


Figure 4.6. A residual block, the building block of ResNet architecture. Taken from [36].

and its own feature-maps are used as inputs for all subsequent layers. This helps to alleviate the vanishing-gradient problem, strengthens feature propagation and encourages feature reuse.

4.3.3 Mixed architectures

Residual blocks, dense blocks and attention modules represent the most important and popular improvements and developments applied to the models, and also to the UNet, in order to build more complex and articulated networks that can improve classification or segmentation accuracy. From the use and combination of these modules, some recent works for medical and non-medical image segmentation were published in the last year. In [38], authors propose a network that includes two modules arranged in a complex fashion in order to better model the inter-dependencies of features in channel and spatial dimensions respectively. The so-called 'channel contextual module' and 'spatial contextual module' are included in the upsampling path and inserted in some dense and compression units, arranged to produce outputs at each depth level, that allow deep supervision of the net. The encoder part of the network is chosen to be the DenseNet-161 [37]. Another interesting work was published in 2020 for polyp segmentation [39], where authors introduce an encoder-decoder architecture with the inclusion of a reverse attention module (RA) and a parallel partial decoder (PPD). The former is an attention mechanism built to recognize discriminative polyp regions through an erasing foreground object manner. The erasing strategy driven by reverse attention can eventually refine the imprecise and coarse estimation into an accurate and complete prediction map. The parallel partial decoder then aggregates high level features with a parallel connection to produce the global segmentation map.

The performance of medical image segmentation has significantly advanced with the convolutional neural networks. However, most existing CNNs-based methods sometimes produce unsatisfactory segmentation mask without accurate object boundaries. To address this problem, authors in [40] formulate a boundary-aware context neural network (BA-Net) for medical image segmentation, to capture richer context and preserve fine spatial information. In each stage of encoder network, pyramid edge extraction (PEE) module is proposed for obtaining edge information with multiple granularities firstly. Then a mini multi-task learning (MTL) module is designed for jointly learning to segment object masks and detect object boundaries. At last, a cross feature fusion (CFF) module aims to selectively aggregate multi-level features from the whole encoder network. By cascaded three modules, richer context and fine-grain features of each stage are encoded.
This study was taken under particular consideration for this work of thesis thanks to its focus on structures boundaries. The contours of knee bones constitute the most critical area indeed, where segmentation errors are most likely to happen. Furthermore, they determine the match between the segmented anatomy and the true bone geometry that is wanted to be as accurate as possible. In this sense, this paper has provided a good starting point for the development of our simpler but yet very effective CEL-UNet.

4.3.4 Knee Joint Bone Segmentation

A recent study performed at Politecnico di Milano and published in 2020 is used as reference for the present thesis work. The study performed by Marzorati et al. [41] is analogue to the present, as it investigates the effectiveness of Unet for tibia and femur segmentation, in patients with severe osteoarthritis for PSI based surgical planning. The dataset was composed by CT scans provided by MEDACTA International SA, even though in a reduced number of case-patients in comparison to the one used in the present study. Results of the former work were very satisfactory and promising, so it was decided to bring the analysis forward by focusing the efforts in exploring the best configurations and enhancing the segmentation quality.
In that previous study, the original dataset was sub-sampled to 4 different resolutions, in order to establish a comparison of segmentation results at different voxel dimensions. Evaluation was done in terms of the surface distance calculation between the predicted segmentation and the ground truths. Results are extremely eloquent, showing a better

segmentation accuracy and a significant lower surface distance for the dataset at higher resolution, as can be seen in figures 4.8. This fact has driven the choice to operate on a new dataset by resampling all the scans at 192×192×192, value that provides a good resolution for training the network.

Name	Dx	Dy	Dz	Δx	Δy	Δz
D1	128	128	128	1.50	1.50	1.00
D2	160	160	128	1.25	1.25	1.00
D3	184	184	160	1.10	1.10	0.80
D4	200	200	192	1.00	1.00	0.70

Figure 4.7. Input volume dimensions and relative pixel spacings used in [41].

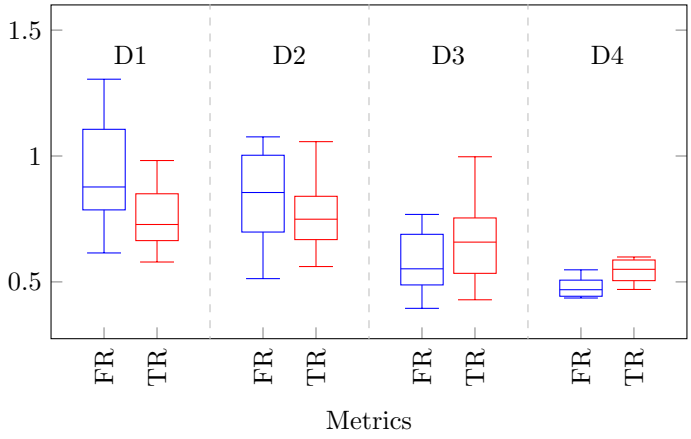


Figure 4.8. Root Mean Squared Errors achieved with Unet at the different resolutions defined in figure 4.7.

Chapter 5

Methodology

5.1 MEDACTA Dataset

This work of thesis arises from the collaboration between Politecnico di Milano and MEDACTA International, with the aim to perform a detailed analysis and research about innovative frameworks for automatic segmentation of the knee bone anatomy, leaveraging deep learning algorithms. A collection of CT images of around 400 patients was provided by MEDACTA international in DICOM standard format, acquired in the context of preoperative planning for total knee arthroplasty intervention. For each patient, some hundreds of axial scans covering the whole knee structure generate the single volumetric data used during training and testing of the algorithms. For effectiveness purposes, the choice was to train 3D models instead of 2D, at the cost of increased computational complexity, training time and memory requirements. A preliminary study about the problem was made, showing that improved results are provided by 3D models with respect to 2D ones. This is also confirmed by the literature in [42] [41], where authors show the great performances of volumetric segmentation, specifically in the case of tube-shaped structures. Every scan that composes each patient's set has a size of 512×512 pixels along x and y directions, with x and y being the components laying in the axial plane. This provides a x-y spatial resolution of 0.39 millimeters for each slice, which allows to precisely discern very thin structures in the two-dimensional slices. Instead, the number of slices for each patient along the z direction, e.g. the longitudinal axis, varies significantly due to different conditions and machines with which scans were acquired and probably also due to different

clinical decisions. It follows that the resolution along z axis is not uniform across all patients and it varies approximately from 0.3 mm to 1 mm. The data is encoded in either 12 or 16 bits, providing grey intensity levels that can span over a wide range of values. Additional important information is stored as metadata in DICOM files, such as the value of the offset location with respect to the origin and parameters defining the relationship between stored values and output units. These numerical values are necessary to perform a few preprocessing steps, which will be explained in detail in the next sections, but they are completely ignored during training and testing of the deep learning models. In figure 5.1, examples of axial, sagittal and frontal slices from the dataset are shown.

Figure 5.1. axial, sagittal and frontal slices extracted from an example CT volume.

During the preliminary analysis of the work, some choices were made to generate a precise definition of the problem and to design the boundaries of this research. Such decisions are summarized below:

- 3D volumetric data was preferred to 2D for effectiveness reasons
- Axial slices were used instead of sagittal or frontal ones
- All osseous structures in the knee were considered for segmentation

Given those consideration, the target structures to be segmented in this work are 5: background, femur, tibia, patella and fibula. This brought to a reduction of the available data for training and testing, since not all the 400 patient folders carried the necessary files to produce the labels for all 4 bone anatomies. In addition, very few data were corrupted and could not be read. In the end, 259 cases were used

and divided into train dataset, validation dataset and test dataset with the following criterion:

Total number of cases: 259			
-	Training	Validation	Test
Percentage	75%	15%	10%
Number	195	39	25

To feed the data into the neural network for training, the 3D input size had to be fixed before the model definition. Also, given that the number of scans along longitudinal axis was not the same for all patients, it was made uniform. In order to fit memory, GPU and training time requirements, all volumes were transformed to a size of 192×192×192. This required some preprocessing, which is going to be fully illustrated in the next sections. The volumes were saved and stored in Nifti format, which includes a .hdr file to store meta-information and a .img file to store the actual image data.

5.2 Preprocessing

Various are the reasons that rendered preprocessing necessary in this work, to take raw dicom data and build a standardized dataset of 3D CT volumes to be used for neural networks training. Dicom files data of CT scans were accompanied by accurate hand-delineated mesh surfaces of knee bone structures stored in .stl files. A scheme of all available data is presented in figure 5.2. Combining knee image data with patients meta information and meshes of target structures, preprocessing was carried out in 4 main steps:

Cropping slices and creating Nifti volumes: to discard useless information

Reshaping volumes: to standardize size of volumes

Creating ground-truth Labels: to generate single-anatomy references

Merging ground-truth Labels: to merge all single-anatomy references in one file

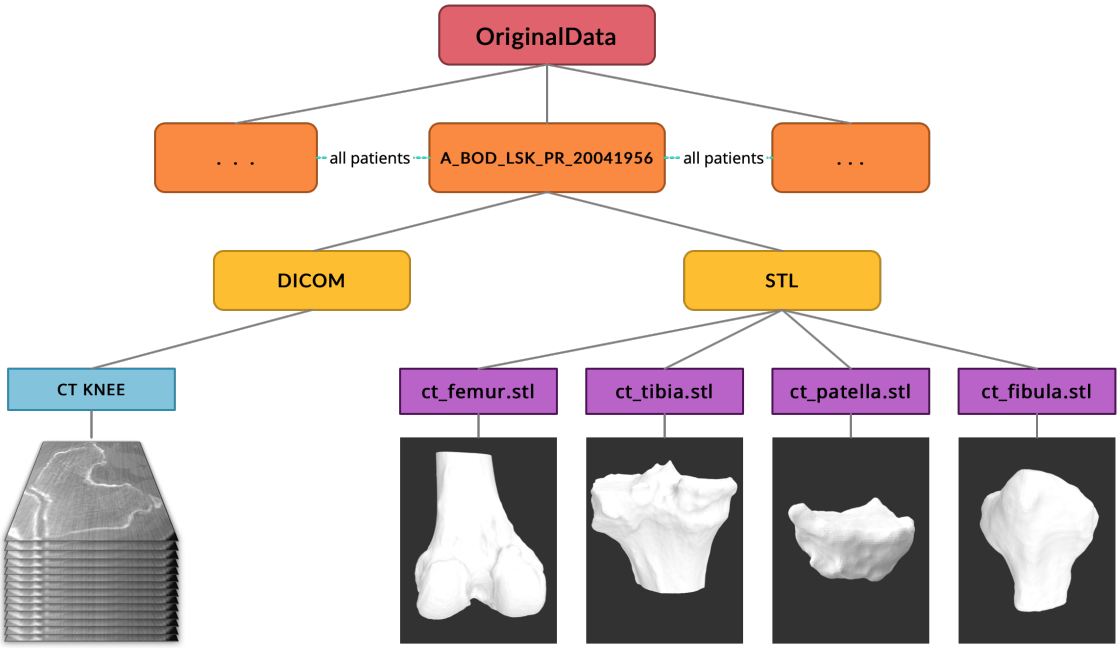


Figure 5.2. Organization of original dataset: a folder for each patient contains dicom files of all axial slices of the knee bones, and a mesh file for each one of the labeled anatomies.

5.2.1 Cropping

Original volumes contain a lot of extra information that is not useful for our purposes and just contributes to increase volume size and therefore computational complexity and training time, causing a significant efficiency loss. Looking at the axial slice in figure 5.1, it is evident how information about target bone anatomies is condensed in the center of the image. The great amount of black and gray pixels, corresponding respectively to the background space and soft tissues, is completely neglectable and therefore it can be discarded by cropping the image at some x and y coordinate values. The same is done along the z direction with slices that are located further from the intra-articular space and just show proximal sections of the femur or distal sections of the tibia. To define the cropping locations, the minimum and maximum coordinates in each of the three axes are extracted from the surface meshes. Using these values, crop is performed, leaving a certain margin between the extreme locations of the anatomies and the final image edges. The volume is then saved as Nifti file. Figure 5.3 shows the results of the described procedure.

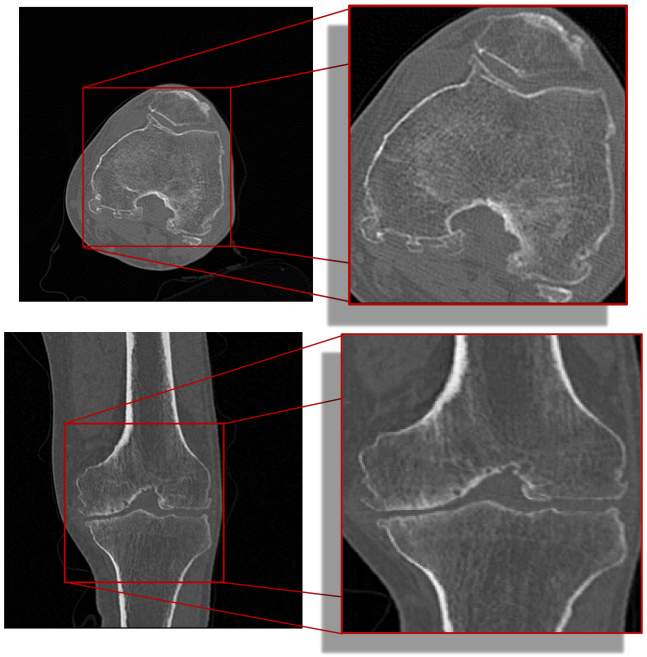


Figure 5.3. Examples of cropping procedure to delete all useless information.

5.2.2 Reshaping

Cropping operation automatically finds the best configuration for each patient, setting the new edges the closest possible to the bones, in order to discard the great amount of useless information about background and soft tissues. It follows that volumes turn out to have different sizes also in x-y coordinates. Reshaping is therefore performed to standardize volume size at 192×192×192 and create a consistent dataset, that can be fed as input of the neural network. After reshaping, a normalization operation is performed to standardize gray values between 0 and 1.

$$V_{norm} = \frac{VR - \min_{i,j,k} VR}{\max_{i,j,k} VR} \tag{5.1}$$

where VR is the volume after reshaping operation and i, j, k represent the three spatial directions.

5.2.3 Labeling

Ground truth of target structures were given as 3-dimensional surface meshes, created from manual segmentations of the CT scans. Starting from this representation, it was created a hard mask that assigns the same numerical value to pixels belonging to the

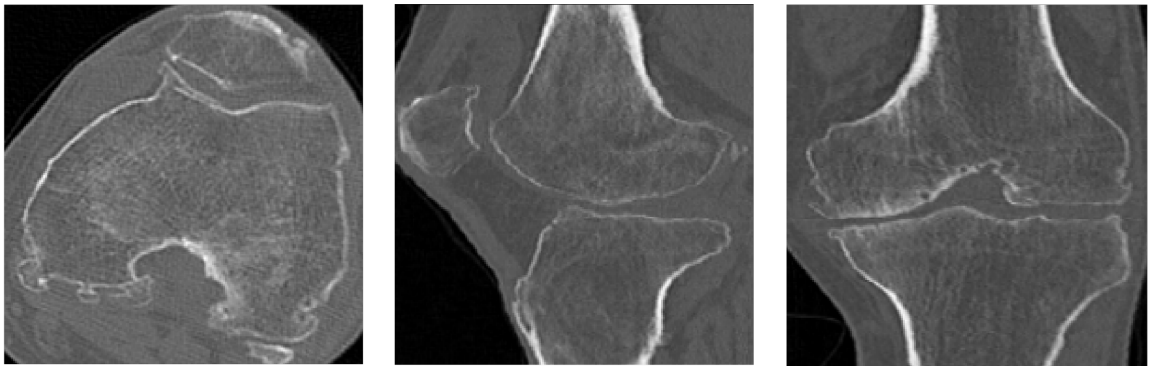


Figure 5.4. Examples of reshaped axial, sagittal and frontal slices.

same bone structure and to the background. For each of the bones, the correspondent surface mesh is sliced at z coordinates extracted from the reshaped volumes. When an intersection is found, contour pixels are registered and set to 1 in an empty volume. These contours are eventually filled to generate the axial binary masks that compose the whole binary volume of the correspondent bone. The same operation is computed for each of the bones. To merge all labels in one, these volumes are summed up, after assigning different integer values for background (0), femur (1), tibia (2), patella (3) and fibula (4). The whole preprocessing was performed with MATLAB but the code

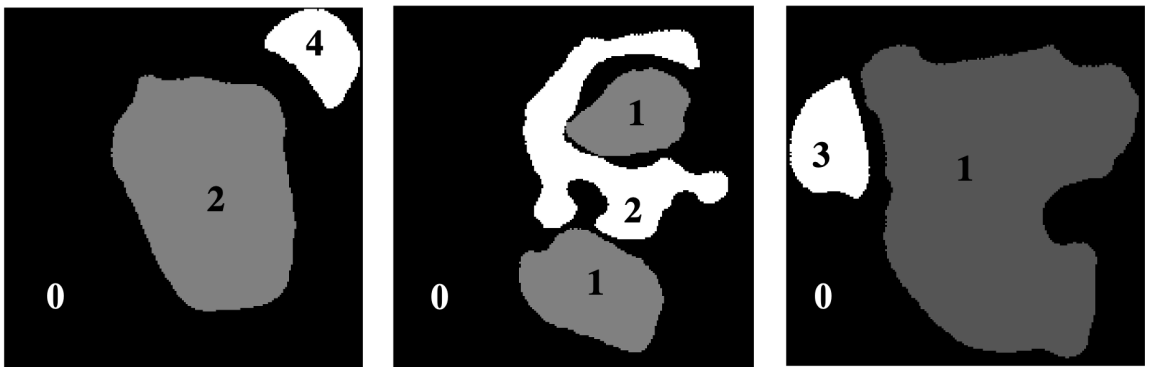


Figure 5.5. Axial slices extracted from a labeled ground truth volume. 0: background, 1: femur, 2: tibia, 3: patella, 4: fibula.

was also converted in Python in order to add it to the segmentation package that is being developed in GitHub. Moreover, it was noted that preprocessing performed with Python code, that leans on popular opensource libraries, is almost 5 times faster that MATLAB computation.

5.3 Workflow

After the deep revision of the literature and the preprocessing of the dataset, the segmentation workflow of this thesis was organized in two main parts. In the first part, the Unet was used as main architecture and efforts were put in finding the most relevant aspects the model should focus on. Since in [41] authors used the dice score as unique loss function for the trained models, it was chosen to address the task by introducing 4 additional loss functions for segmentation, taken from the literature. A comparison was set up to find the most suitable one for the proposed problem, and to understand how different functions influence the network training.
The second part of the work took advantage from results observed in the first one. An innovative encoding-decoding architecture that takes its roots from the Unet was designed and developed with the aim of enhancing the performances of the automatic segmentation. The network was called CEL-Unet, from the name of the targeted loss function chosen for it: Combined Edge Loss (CEL) function.

5.4 Proposed Loss Functions

5 different loss functions were used to train 5 models that share the same architecture, the Unet. Each one of these functions is going to be explained in detail in the next paragraphs.

5.4.1 Weighted Dice Loss

The dice loss is one of the most used objective functions for segmentation tasks. In this work, this can be referred to as a soft dice index, as it is computed using the soft probability mask produced by the output layer of the network. Given that classes are not equally represented in the dataset, in particular for patella and fibula, the computed dice index evaluates differently the contributions of each class, thanks to some pre-computed weights:

$$w_c = \frac{\sqrt{\frac{1}{N_c}}}{\sum_{i=1}^{C} \sqrt{\frac{1}{N_i}}} \tag{5.2}$$

w_c is the weight factor for class c, N_c is the number of voxels belonging to class c and

the sum at the denominator runs through all the C classes. This helps not to bias the discriminator during training, which otherwise would tend to predict the most frequent class for all voxels. Weighted Dice formula can be written as:

$$\mathcal{D}(y, \hat{y}) = \sum_{c=1}^{C} w_c \left(\frac{2 \cdot \sum^{N} y_c \cdot \hat{y}_c}{\sum^{N} y_c \cdot y_c + \sum^{N} \hat{y}_c \cdot \hat{y}_c}\right) \tag{5.3}$$

where y_c and $\hat{y}_c$ are respectively the true and predicted probability of segmented volumes for the label c whose scalar product is computed over N voxels. The corresponding loss function is simply written as:

$$\mathcal{L}_{dice} = 1 - \mathcal{D}(y, \hat{y}) \tag{5.4}$$

5.4.2 Focal Loss

Focal loss is introduced in 2018 by [43] as a loss function to address the problem of foreground-background class imbalance. The standard cross entropy loss is here reshaped such that it down-weights the well-classified examples, increasing instead the penalty assigned to more difficult ones. This loss function focuses training on a set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training.

$$\mathcal{L}_{focal} = -\frac{1}{N} \sum_{c=1}^{C} \left(\sum^{N} (1 - \hat{y}_c)^{\gamma} \cdot y_c \cdot \log \hat{y}_c\right) \tag{5.5}$$

The term $(1 - \hat{y}_c)^{\gamma}$ is the focal factor that assigns a weight to each voxel. The higher the classification confidence of a voxel, the lower will be its weight in the function. γ is a hyperparameter experimentally set to the value of 2. Since the extent of the class imbalance is not extreme in our application, it has been chosen to use this function in combination with the dice index, so to help and fasten the convergence of the optimization algorithm, as reported in [44]. A parameter α was used to weight the contributions of the two components in the final loss, and a scheduling strategy was set up to dynamically change the weights of both during training.

$$\mathcal{L}_{out} = \alpha \cdot \mathcal{L}_{dice} + (1 - \alpha) \cdot \mathcal{L}_{focal} \tag{5.6}$$

The parameter α was set to 1 at the beginning and its value was decreased of 0.005 for each epoch of training, until the value of 0.3. In this way, dice coefficient helps the convergence at the beginning of the training and focal loss enters progressively to refine the segmentation for hard examples.

5.4.3 Exponential Logarithmic Loss

This loss function is proposed by [45] and it is specifically introduced to enhance segmentation of small objects. Even if nor tibia neither femur can be considered as small structures, this loss function was included because of the non-linearity introduced by the logarithmic transformation, which can indeed act as a focal factor, increasing attention given to misclassified structures. Two contributions are computed:

$$\mathcal{L}_D = \sum_{c=1}^{C} w_c(-\log \mathcal{D}_c(y, \hat{y}))^{\gamma_D} \tag{5.7}$$

$$\mathcal{L}_{CE} = \sum_{c=1}^{C} w_c(-\frac{1}{N}\sum^{N}(y_c \cdot \log \hat{y}_c)^{\gamma_{CE}}) \tag{5.8}$$

As it can be noted in the formulas, the two contributions come from a logarithmic and exponential transformations of dice coefficient ($\mathcal{D}_c(y, \hat{y})$) and an exponential transformation of the cross entropy function. The two gamma values, γ_D and γ_{CE}, are parameters introduced to further adjust the non-linearity given by the logarithms. Experimental values of both are set to 1. C indicates the number of classes and w_c is the weight assigned to each class in the cross entropy function, useful to face class imbalance problem and it is computed in the same manner as it was for the Weighted Dice Loss function. The two quantities are merged together in the final loss with a parameter β, which is experimentally set to 0.8.

$$\mathcal{L}_{out} = \beta \cdot \mathcal{L}_D + (1 - \beta) \cdot \mathcal{L}_{CE} \tag{5.9}$$

5.4.4 Double Cross Entropy Loss

This loss is a development of the simple cross entropy loss that it is usually found for segmentation tasks. It has been called double cross entropy, because the contribution to the final loss is given by two cross entropy terms, the first defined on the soft probability mask and the second defined on the inverse of it. This can help the segmentation of structures with sparser representation.

$$\mathcal{L}_{CE1}^{(c)} = -\frac{1}{N}\sum^{N}(y_c \cdot \log \hat{y}_c) \tag{5.10}$$

$$\mathcal{L}_{CE2}^{(c)} = -\frac{1}{N}\sum^{N}((1 - y_c) \cdot \log (1 - \hat{y}_c)) \tag{5.11}$$

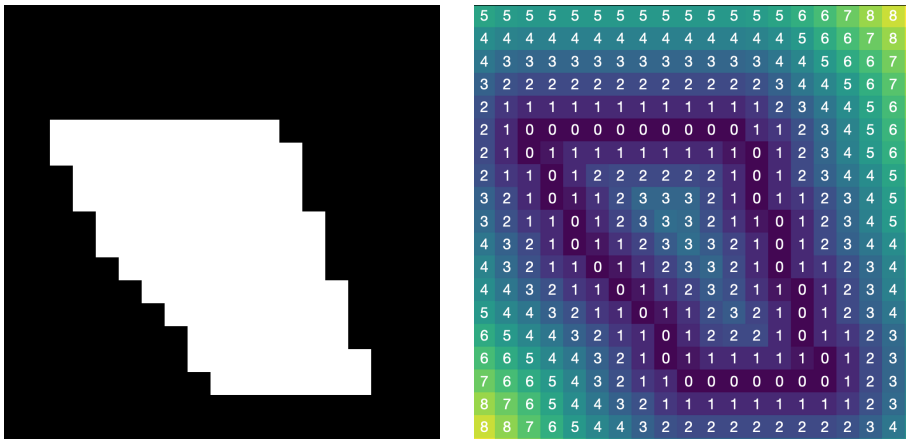
The two contributions are equally balanced and summed together, after each class term is multiplied for its weight that accounts for class imbalance. Final formulation of the output loss is the following:

$$\mathcal{L}_{out} = \sum_{c=1}^{C} w_c * (0.5 \cdot \mathcal{L}_{CE1}^{(c)} + 0.5 \cdot \mathcal{L}_{CE2}^{(c)}) \tag{5.12}$$

5.4.5 Distanced Cross Entropy loss

The last objective function introduced in this comparison is still cross entropy-based. The idea behind the Distanced Cross Entropy comes from the observation that most of the segmentation errors made by the network are found to be along the boundaries of the anatomies. Here, the bones show deformations and irregularities, in particular in proximity of the intra-articular space. Also, the presence of evident osteophytes around the anatomies usually fools the model that tends to include them in the bone segmentation, even if they should be sometimes ignored during preoperative planning and manufacturing of the cutting guides.
These considerations led to the introduction of a loss function that specifically focuses on boundary voxels, with the aim of increasing the accuracy of segmentation and decrease the mean surface distance error. The approach followed was the one presented in [46], where a distance map is created to weight more the voxels closer to boundaries

and assign lower importance to further ones, that are quite always correctly classified. The distance map is created starting from the Euclidean Distance Transform (EDT), which assigns to each voxel the value of its distance from the closest voxel belonging to the boundary of the target structure. Figure 5.6 shows an example object on the



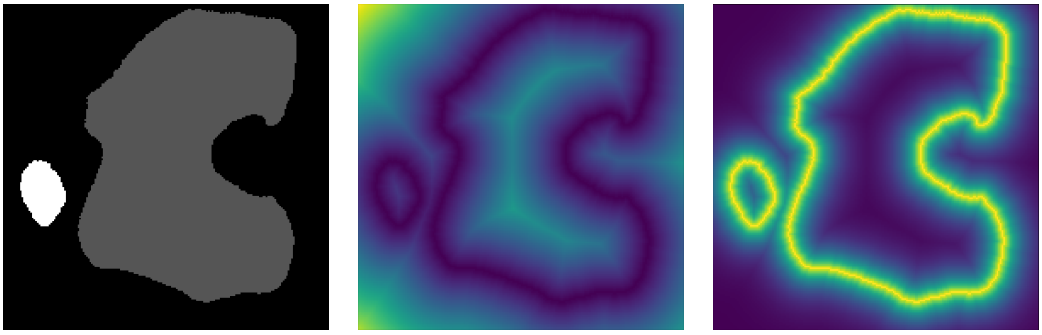
(a) Example object (b) Euclidean Distance Transform

Figure 5.6. Image showing an example binary structure on the left, and its Euclidean Distance Transform on the right. Pixel values are printed on top of the image and indicate rounded distances from structure boundaries.

left and its Euclidean Distance Transform on the right, on which pixel values of the EDT are printed. The EDT allows to recognize boundary voxels, which are given the value of 0. Now, the so-called Distance Weight Map (DWM), that increases the values of boundary voxels, can be computed, by using a negative exponential transformation applied to the EDT. Figure 5.7 shows, from left to right, the ground truth mask for a random axial slice inside the volumetric data of a patient, its EDT and the corresponding Distance Weight Map. Brighter pixels towards yellow are representative of higher values, while darker and blueish ones represents lower values. The DWM is then multiplicated to the classic cross entropy function, balancing in this way the importance of edge voxels against all the others. Mathematical expression of the loss function is shown below.

$$\mathcal{L}_{Dce} = -\frac{1}{N}\sum^{N}(\sum_{c=1}^{C} DWM_c \cdot y_c \cdot \log \hat{y}_c) \tag{5.13}$$

$$DWM_c = 1 + \gamma * exp(-\frac{EDT_c}{\sigma}) \tag{5.14}$$



(a) Ground Truth Mask (b) EDT (c) DWM

Figure 5.7. Image shows an axial slice with femur and patella anatomies (a), its EDT (b) and the exponential transformation of the EDT (DWM) that assigns higher weights to voxels closer to the boundaries (c).

The DWM is computed with the negative exponential transformation, with the values of γ and σ set to 8 and 10 respectively, following the approach presented in [46]. Distanced Cross Entropy alone would not guarantee an easy convergence of the optimization algorithm, because of the bias given to the loss by the much greater importance of boundary pixels. This would often lead to the generation of segmentation masks with some holes inside, due to the fact that pixels in the core of the bones are further away from the edges, so they are given less importance and hence can be more easily misclassified. For this reason, as it was done for the Focal Loss, the weighted Dice coefficient was maintained as strong contribution inside the final loss, and weighted by a factor α, which takes a value in [0,1]. In particular, the initial value of α was set to 1 and it was decremented of 0.05 every 40 epochs, over a total of 150. In this way, inside every interval of epochs, the optimizer was forced to look for the minimum of a slightly different function, progressively more focused on critical border areas. Final loss can be written as follows.

$$\mathcal{L}_{out} = \alpha \cdot \mathcal{L}_{dice} + (1 - \alpha) \cdot \mathcal{L}_{Dce} \tag{5.15}$$

5.5 Proposed Architecture: CEL-Unet

The second part of this work of thesis was dedicated to the search and the development of a new, innovative and targeted model that could overcome Unet performances and constitute an interesting alternative approach for other studies and researches. The

idea was to maintain the backbone structure of the encoding-decoding architecture, which, for our knowledge, still remains the most effective CNN for segmentation, and improve it with the addition of simple but effective modifications. The good and promising results provided by the Unet trained with the Distance Cross Entropy Loss proved the potential improvement that could be gained by focusing also on the edges, instead of just on the whole anatomies. This has been the core idea for the conception of the CEL-Unet, that took the name from the loss function introduced and utilized for the model, the Combined Edge Loss (CEL) function.
The CEL-Unet is a network with the same encoder-decoder structure introduced by the Unet. While the down-sampling encoding path remains analogue to the former, the up-sampling path represents the main innovation in this new model. Based on the greater consideration that was given to bone boundaries in the analysis and preliminary observations, this approach introduces a dedicated path for the specific learning of interested bone contours. Starting from the deepest level in the network (bottleneck), an additional decoding path, parallel to the original one, is added with the aim of segmenting regions' boundaries. Therefore, two parallel decoders (Mask and Edge) inside the network produce two outputs for each input, one representing the classical semantic segmentation mask and the second representing a semantic boundary detection map. Skip connections from the encoding branch are maintained and now directed to both decoding branches, Edge and Mask. Since the final segmentation is required to be a map representing the filled structures, the output of the region-aware branch is taken as the ultimate output of the network. To incorporate the new accurate boundary information encoded in the boundary-aware branch though, some unidirectional connections are included in the architecture, to link the two parallel branches. At two depth levels, feature maps produced in the Edge branch are aggregated with feature maps inside the Mask branch. The result of the convolution is forwarded inside the Mask branch to further processing until it reaches the output. CEL-Unet structure is shown in figure 5.8.

5.5.1 PEE Module

Inside the upsampling Edge branch, the network learns to produce thin boundary predictions. However, the boundary of the bones is usually complex and diverse. Hence, in order to obtain a robust boundary information supplement, the model was strengthen by employing a simple and effective pyramid feature extraction scheme for

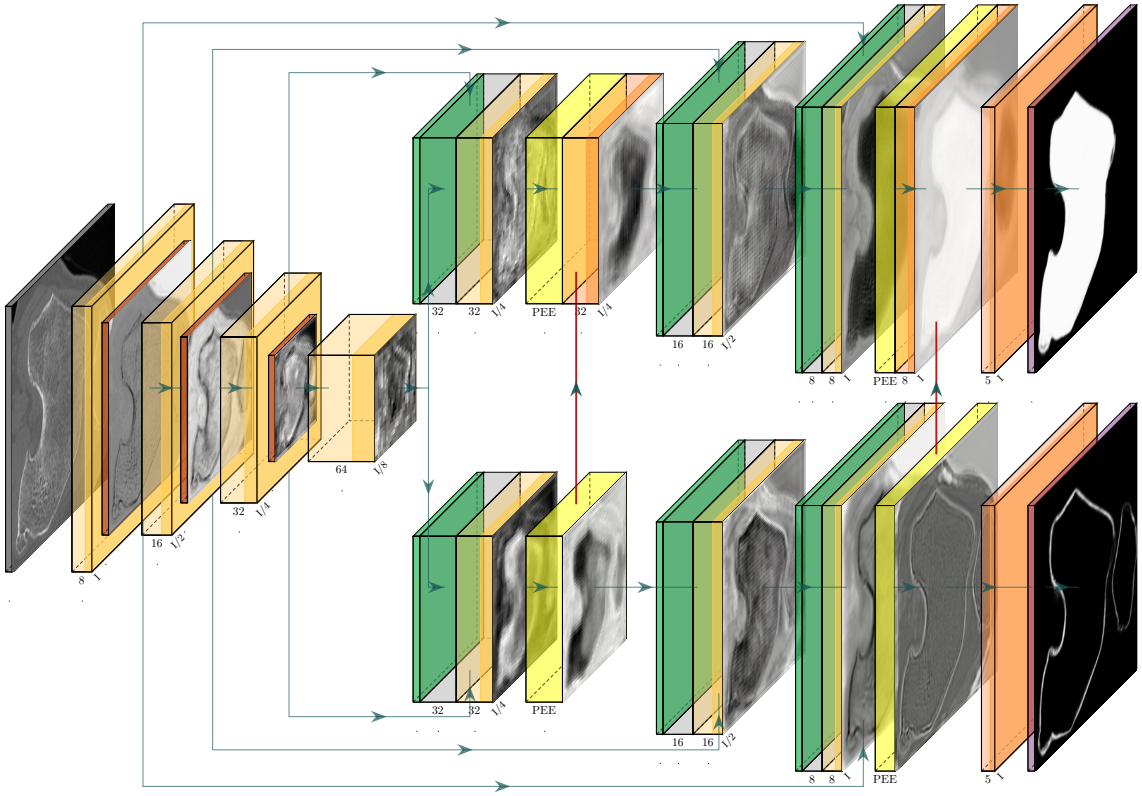


Figure 5.8. CEL-Unet architecture. The double decoding path allows to extract robust and explicit boundary information, that is integrated into the main mask branch through the red vertical connections. Classic horizontal skip connections from down-sampling towards both up-sampling branches are maintained.

mining multi-granularity edge features, inserted in two stages of the decoder Edge branch. This is called Pyramid Edge Extraction (PEE) module and is taken from [40]. Multiple granularities edge features are obtained by subtracting the value of average pooling, performed with different kernel sizes, from its local convolutional feature maps. The core operations of this module are presented in the following formulas. With i denoting the current up-sampling stage, a first 1×1×1 convolution is performed to squeeze the P_i feature maps of current tensor F_i in half, producing F_i', which enters the PEE module.

$$F_i^{(s)} = F_i' - avg_s F_i', \qquad s \in \{1, 2\} \tag{5.16}$$

$F_i^{(s)}$ denotes the edge features of current ith stage with the sth pooling operation, and avg_s is the corresponding average pooling operation. In order to integrate the obtained pyramid edge features, we aggregate them with the features of current stage with a concatenation, and merge them using a 1×1×1 depth convolution operation

denoted by $\mathscr{F}$, that recovers the initial number of feature maps P_i:

$$F_i^P = \mathscr{F}(\mathcal{C}(F_i^{(1)}, F_i^{(2)}, F_i'), P_i) \tag{5.17}$$

In equation 5.17, $\mathcal{C}$ refers to the concatenation process. F_i^P is the output feature maps of PEE module at current stage of decoding Edge branch. By extracting and integrating boundary information with different granularities, the edge features are effectively improved and noise is suppressed. PEE module is graphically represented in figure 5.9.

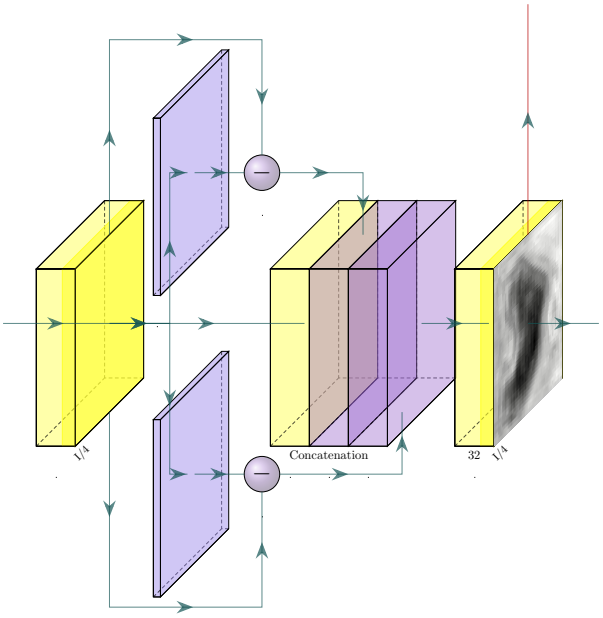


Figure 5.9. PEE module. The yellow box on the left represents F_i'. Two different average pooled tensors are subtracted to F_i', to produce $F_i^{(1)}$ and $F_i^{(2)}$. All tensors are concatenated and subjected to a final convolution.

5.5.2 CEL: Combined Edge Loss

Since the network produces two outputs, the loss function that was chosen for it comprehends two different contributions, one for Edge output and one for Mask output. For the Mask output, the choice was to use the function that best performed on the Unet in the comparison study made in the first part of this work. Therefore, the Mask Loss is defined as the Distanced Cross Entropy Loss, which intrinsically already includes strong focus on regions' boundaries, thanks to the Distance Weight Map used to weight each voxel's contribution.

$$\mathcal{L}_{mask} = \alpha \cdot \mathcal{L}_{dice} + (1 - \alpha) \cdot \mathcal{L}_{Dce} \tag{5.18}$$

Also in this case, a parameter α is used to weight the contributions of the distanced cross entropy and the dice coefficient, but it was assigned a slightly different scheduling policy. Its initial value was set to 0.9 and was decreased of 0.005 each epoch of training, until the minimum value of 0.4. Regarding the Edge output instead, we had to consider the huge imbalance between the number of target contour voxels and all other background voxels. For a random example volume inside the dataset, it is found that femur and tibia boundary voxels are respectively in a ratio of around 0.0072 and 0.0047 with respect of all other background voxels. This difference gets even more enhanced when patella and fibula are considered, since they are smaller bones and include much fewer voxels. For this reason, for the Edge Loss we chose the class-balanced cross entropy function, presented before as the Double Cross Entropy, that allows to alleviate the impact of the higher missing rate that characterizes semantic boundary detection. This loss was modified for the present task, with the inclusion of a multiplication to the Distance Weight Map, in order to further force the model to focus on boundary voxels. Formula of the Edge Loss is reported below:

$$\mathcal{L}_1^{(c)} = \frac{1}{N}\sum^{N}(-DWM \cdot y_c \cdot \log \hat{y_c}) \tag{5.19}$$

$$\mathcal{L}_2^{(c)} = \frac{1}{N}\sum^{N}(-DWM \cdot (1 - y_c) \cdot \log (1 - \hat{y_c})) \tag{5.20}$$

$$\mathcal{L}_{edge} = \sum_{c=1}^{C} w_c * [\beta \cdot \mathcal{L}_1^{(c)} + (1 - \beta) * \mathcal{L}_2^{(c)}] \tag{5.21}$$

The DWM used here is the same as the one used in the Mask loss, with γ and σ parameters taking the values of 8 and 10 respectively. β parameter in equation 5.21, instead, is a weight factor that helps the balancing between the few positive boundary voxels and the great number of negative background ones. Its value is computed as the ratio between the number of boundary voxels and the total number of voxels in the volume:

$$\beta = \frac{N_{edge}}{N_{total}} \tag{5.22}$$

5.6 Implementation Details

All models trained on Unet architecture have a depth of 5 levels. Each time the network goes one step deeper, the number of filters in the convolutions is doubled and the dimension is halved. Every model starts with a first double convolution computed using 8 filters, and reaches the number of 16, 32, 64, 128 and 256 filters for the convolutions computed in each of the subsequent levels, included the bottleneck. Regarding the down-sampling branch, filters used in the convolutions are 3 dimensional, as the input volumes, and their size is (3,3,3). They are applied with a stride of 1 and a padding operation that allows output feature maps to be of the same size as the input. Pooling operation instead uses kernel of size (2,2,2) with a stride of 2 in every dimension, ensuring the down-sampling of input volumes of a factor 2. Batch normalization and L2 regularization is then used.
For what concerns up-sampling branch, similar choices were made: transposed convolutions are performed in each level, with kernels of size (2,2,2) that span over the input with stride of 2 in every dimension, followed by two consecutive convolutions. This enables to increase volume size of a factor 2 along rows, columns and slices, again, so that at the end of up-sampling branch, the output of the model will be produced with the same size of the input.
Down-sampling branch of the CEL-Unet is set in the same way, except for the depth, which reaches a maximum of 3 levels. Initial number of filters for the convolution is 8, which is then increased to 16, 32 and 64 lastly, in the deepest level. Decoding branch is different, with the division into the two parallel branches and the introduction of PEE-modules in the Edge branch. PEE modules include a first initial (1,1,1) convolution, performed to half the number of feature maps. Kernel sizes of the two average pooling operations are set to (3,3,3) and (5,5,5) for the second depth level, where tensors are smaller, and to (5,5,5) and (7,7,7) in the zero level, where feature maps have recovered the original input size. After the two subtractions and the concatenation presented in equations 5.16 and 5.17, an additional (1,1,1) convolution is performed to recover initial number of feature maps. As output layer, for all architectures a softmask layer is used, to compute class scores for final segmentation maps.
For each trained model the Adam optimizer [47] was used with a learning rate set to 3e-5. Number of epochs was set to a maximum of 200 for each model, but training was manually stopped if no improvement was observed for some consecutive iterations. All models were trained using mini-batch approach, each one with a size of 2. Bigger batches could not be set because of the limited memory available. Table 5.1

summarizes all training details.

Training Details			
LR=3e-5	BS=2	$Opt.$=Adam	$Epochs_{max}$=200
		Unet	**CEL-Unet**
Conv. size [stride]		(3,3,3) [1]	(3,3,3) [1]
Deconv. size [stride]		(2,2,2) [2]	(2,2,2) [2]
Pooling size [stride]		(2,2,2) [2]	(2,2,2) [2]
Depth		5	3
N. of filters		8-16-32-64-128-256	8-16-32-64

Table 5.1. Summary of the training details.

5.7 Frameworks and Data handling

All models were developed using Keras API running on top of Tensorflow libraries. Keras is a Python package for building, training and evaluating neural networks, which enables to speed up the development by providing a high-level API that allows to focus more on the network architecture and parameters rather than on implementation. The whole work was done on Google Colab, a free environment to write and run Python scripts leveraging high performance Google hardware, including GPUs and TPUs. Codes run on Google servers, nothing is required except for a laptop, an internet connection and a browser. However, resources are limited in terms of computational power, available memory and usage time. Limited available memory constitutes a serious problem when training three dimensional models, therefore a mixed policy for storing tensors of the model during training was chosen.
Today, most models use the float32 dtype, but modern accelerators can sometimes run operations faster in the 16-bit dtypes. They have specialized hardware to run 16-bit computations and 16-bit dtypes can be read from memory faster. There is no clue about the hardware Google Colab will provide when initializing the Notebook and this means that not always the use of float16 dtype will result in faster training. However, this approach is very useful to reduce the memory requirements of the tensors during training, also because it ensures that training quality will not be affected. Tensorflow

library recently developed an implementation of a mixed type policy for training deep learning models. This was found very practical and was used in this work to train CEL-Unet model, which otherwise would have caused memory overflow with the free resources available.

Chapter 6

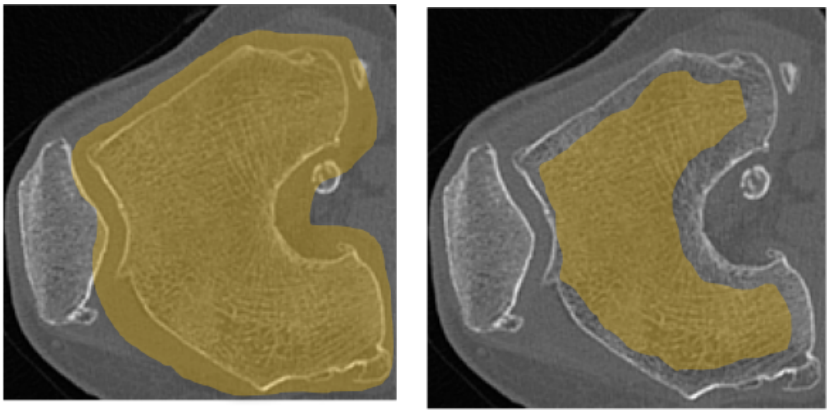
Results

This chapter illustrates results of this work of thesis. All metrics used to evaluate segmentation results are reported in the first place. All different loss functions are compared based on the presented metrics, and the same is done for the newly introduced CEL-Unet. Since the training of 3D segmentation models is expensive in terms of time and resources, a brief comparison between training timings obtained with the different loss functions and different models is provided. A statistical test is performed on the results, to assess significant differences between the experiments.
In the second place, a back analysis on the test set is done, with the aim of visually inspecting and comparing results obtained by each test. Furthermore, this analysis helps to understand where the models could be strengthened and made more robust.

6.1 Metrics: over- and under-segmentation

6.1.1 Volumetric assessment

Segmentation results were evaluated on the test set with targeted metrics, in order to access accuracy and reliability of the trained models. No unique metric can be enough to provide the complete scene of the segmentation results, hence a combination of 3 main metrics was used to access global volumetric accuracy of models. Mainly, 2 are the cases when the network fails: over-segmentation and under-segmentation, as figure 6.1 shows.



(a) (b)

Figure 6.1. Over-segmentation (a) and under-segmentation (b) errors shown on a 2D axial slice.

In the former case, the network includes in the bone prediction voxels that belong to the soft tissues (considered as background in our context), adding False Positives. In the latter case, the opposite happens, with the addition of more False Negatives. As a global indicator of the label and prediction level of overlapping, the Jaccard Coefficient is used, with the formulation given in equation 6.1. It computes the intersection over union between two segmentation masks. It is directly related to the Dice coefficient, so monitoring both Dice and Jaccard does not provide any additional information.

$$Jaccard = \frac{TP}{TP + FP + FN} \tag{6.1}$$

This metric gives a snapshot of the accuracy provided by the model, as it accounts for all misclassified voxels, positives and negatives. However, information about which error happens the most is lost. For this reason, two additional metrics are considered, the first sensitive to over-segmentation, while the second to under-segmentation:

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

All metrics take values in [0,1] and were computed counting TP, TN, FP and FN for each class. Considering that the background includes the higher number of voxels

among all classes, it is intuitive that this class will more easily reach values close to 1 in all metrics. In a similar way, given that femur and tibia are bigger than patella and fibula, they will reach higher scores, when the same number of falsely classified voxels is counted.

6.1.2 Surface assessment

Together with the volumetric analysis of the segmentation, a surface similarity assessment was carried out, to evaluate the distance between the reference and the reconstructed anatomies. Two distance measures were considered: the Hausdorff distance and the root mean square error. To compute these quantities, three dimensional meshes for each bone were extracted from the output volumes, using marching cubes algorithm. In this way, the coordinates of the vertices belonging to the reconstructed surfaces were compared to the ones of the meshes provided in the original dataset, to allow surface distance assessment.
Hausdorff distance is defined as the greatest of all the distances from a point in one set to the closest point in the other set. Hence, it represents the maximum error made for each surface. Root mean squared error instead, as the name suggests, computes an average of all the distance values from vertices of the first surface to closest vertices of the second one. In this work, all distance values are computed starting from the points belonging to the reference surface and going towards points of the reconstructed one, bringing to the following formulations:

$$H(X,Y) = \max_{x \in X}\{\min_{y \in Y}\{d(x,y)\}\} \tag{6.4}$$

$$RMSE(X,Y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\min_{y \in Y}\{d(x_i,y)^2\}} \tag{6.5}$$

where X is the set of the N reference vertices and Y the set of reconstructed vertices.

6.1.3 Statistical Analysis

A statistical analysis was performed for the Jaccard index obtained in the volumetric evaluation, in order to asses significant statistical differences between results distributions for each test. The analysis was performed for both global and local evaluations, hence including femur, tibia, left condyle, right condyle, femur trochlea and tibial plateau. For each of these anatomies or regions, a Kruskal-Wallis test was carried out to understand if distributions of the Jaccard results were statistically different, using a confidence level of 95%. Given that all statistic tests gave significant results, a post-hoc comparison was also performed, with the aim of understanding where the statistical differences truly came from, and so to compare the different tests with each other.

6.2 Test Set

All trained models were evaluated on a test set of 25 cases, randomly extracted from the initial available dataset and never seen by the models. All volumes were labeled by experts, hence the ground truth segmentations of femur and tibia can be considered very accurate. Nevertheless, near the end of the work we were informed that, during labeling, less attention was paid to patella and fibula anatomies. This happened because these two bones are not clinically considered during PSI-based TKR. They were included in the knee labels just for visual reasons, basically to provide a more realistic representation of the whole anatomy. Anyway, this fact did not impact on the training process, which was successful for all 4 anatomies. However, it tended to bias the results on the test, since some well-reconstructed shapes of patella and fibula were compared and overlapped to some less faithful reference labels, which brought to a degradation of the metric scores. Segmentation results about patella and fibula will be reported in the next sections as well, but they have not to be considered of clinical relevance, for the mentioned reasons.

6.3 Experimental Results

Analysis of segmentation results was performed in two separate steps. Firstly, a global segmentation assessment has been performed, in which full segmented anatomies were compared to ground truth volumes. This evaluation estimates the goodness of segmentation models. However, large areas of both femur and tibia anatomies are tube-shaped, which make them easy to segment. Furthermore, these areas are not crucial for jigs manufacturing, since they are far from the knee joint. As shown in figure 2.1 in chapter 2 indeed, the most relevant osseous surfaces considered in preoperative planning are the ones facing the knee joint: tibial plateau, femur condyle and femur trochlea. Here, the cutting guides need to be safely hooked to allow precise bone resections. Hence, a localized segmentation assessment in these areas was also carried out, in order to understand how well the reconstructed surfaces match the real ones.

6.3.1 Global Results

Results of the loss functions comparison are here shown and discussed, together with results obtained with the CEL-Unet. As previously exposed, 5 loss functions were chosen to train 5 models that leverage the Unet architecture: Dice Loss, Focal Loss, Exponential-Logarithmic loss, Double Cross Entropy Loss and Distanced Cross Entropy loss. While the dice is surely the most used for overlapping assessment, and can be considered as a baseline, the other 4 losses were found in the literature, and each one stresses a peculiar aspect during training. The comparison was established in order to understand from which of the losses the training process could benefit the most. CEL-Unet model was trained using the Combined Edge Loss function instead, accounting for the two outputs of the model. Tables 6.1, 6.2, 6.3, 6.4 show the scores obtained by the different models across all 4 anatomies. Given the considerations previously made in section 6.2 regarding patella and fibula, the analysis will focus just on results achieved on femur and tibia, that are clinically relevant. All results are shown anyway, for completeness reasons.

The values in the tables indicate the mean across data in the test set for each metric. Bold numbers highlight the best score recorded for each metric, considering

FEMUR

	Jaccard	**Recall**	**Precision**
Dice	0.9412	0.9628	0.9767
Focal	0.9432	0.9511	**0.9912**
Exp Log	0.9317	**0.9823**	0.9475
Double CE	0.9134	0.9751	0.9349
Distanced CE	**0.9592**	0.9812	0.9770
CEL-Unet	**0.9666**	**0.9871**	0.9790

Table 6.1. Femur results.

TIBIA

	Jaccard	**Recall**	**Precision**
Dice	0.9433	0.9757	0.9659
Focal	0.9475	0.9668	0.9792
Exp Log	0.9405	0.9773	0.9615
Double CE	0.9007	**0.9793**	0.9182
Distanced CE	**0.9493**	0.9679	**0.9800**
CEL-Unet	**0.9576**	**0.9814**	0.9753

Table 6.2. Tibia results.

PATELLA

	Jaccard	**Recall**	**Precision**
Dice	0.8863	0.9467	0.9337
Focal	0.8796	0.9281	0.9449
Exp Log	0.8719	**0.9516**	0.9132
Double CE	0.8667	0.9356	0.9239
Distanced CE	**0.8892**	0.9371	**0.9469**
CEL-Unet	**0.8953**	**0.9539**	0.9369

Table 6.3. Patella results.

just models leveraging Unet, with the 5 loss functions. Instead, CEL-Unet results are written in bold if they overcome the best value obtained with the simple Unet. Since a single metric value does not suffice for a complete explanation of model's performances, all values must be analyzed at the same time.

FIBULA

	Jaccard	**Recall**	**Precision**
Dice	**0.8066**	**0.8996**	**0.9337**
Focal	0.7728	0.8336	0.9267
Exp Log	0.7995	0.8871	0.8979
Double CE	0.6923	0.8457	0.8011
Distanced CE	0.8059	0.8991	0.8917
CEL-Unet	0.7655	0.8118	**0.9414**

Table 6.4. Fibula results.

An initial comparison of the 5 losses with the Unet architecture is done. The higher score obtained for tibia and femur with the Distanced Cross Entropy in the Jaccard index suggests that this loss function globally achieves the most accurate segmentation. This value also accounts for the ability of each model to maintain an unbiased behavior towards over- and under-segmentation, since it consider at once both false positives and false negatives. Actually, there are cases in which the model achieves a very high score on precision and a significantly lower one on recall, and viceversa, denoting an unbalanced segmentation. This happens for the Focal loss and Exp Log loss for femur, or the Double Cross Entropy Loss for Tibia. As a matter of fact, despite the Focal loss achieves the second best score in the Jaccard index for femur and tibia, it generally tends to under-segment structures. In a similar way, Double Cross Entropy and Exponential Logarithmic losses seem to be more biased towards the addition of False Positives, meaning over-segmentation. On the contrary Dice and Distanced Cross Entropy losses achieved more balanced results on these metrics.
CEL-Unet model was able to outperform the former ones in almost all aspects. Jaccard coefficient increases of almost 1 percentage point in both femur and tibia, and both precision and recall record high values for the two principal anatomies. Boxplots are also reported in figures 6.2, 6.3, 6.4, 6.5 to provide a more general overview of results, where it is easy to see how models behave towards over- or under-segmentation.

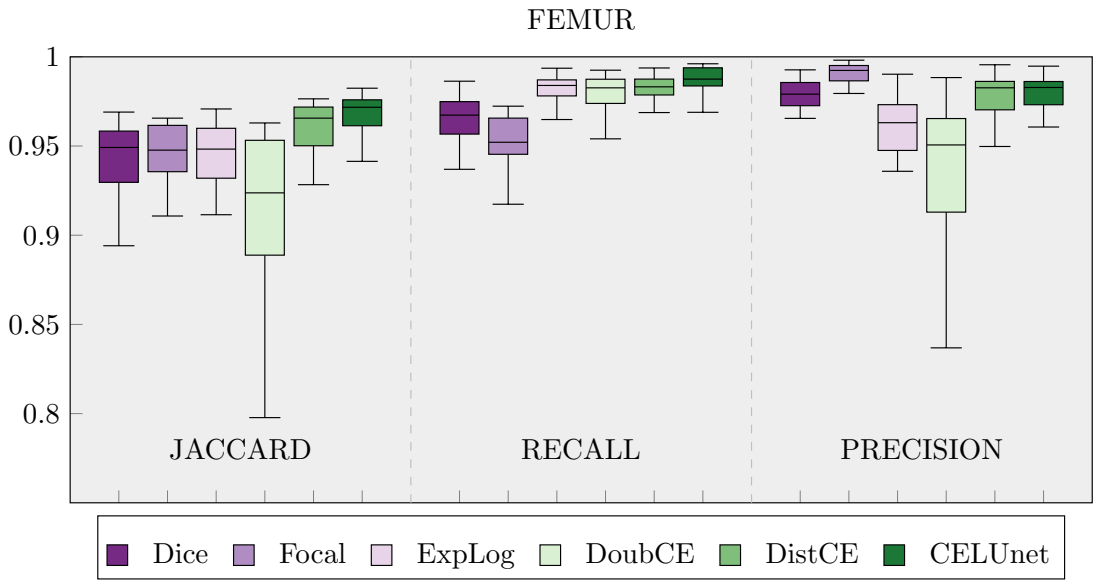


Figure 6.2. Boxplot of Jaccard, Recall and Precision score distributions for segmented femur.

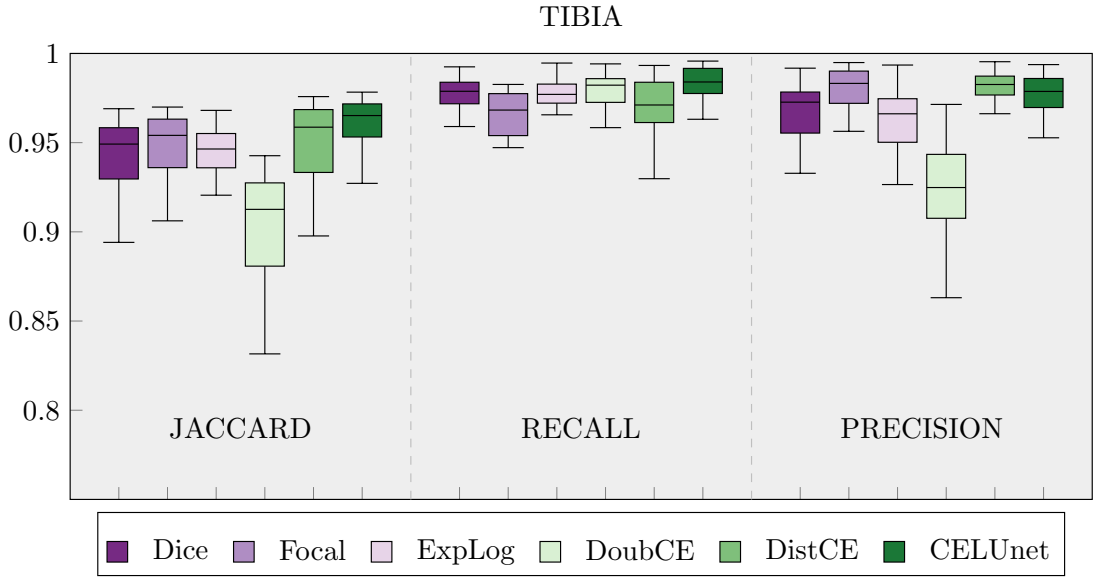


Figure 6.3. Boxplot of Jaccard, Recall and Precision score distributions for segmented tibia.

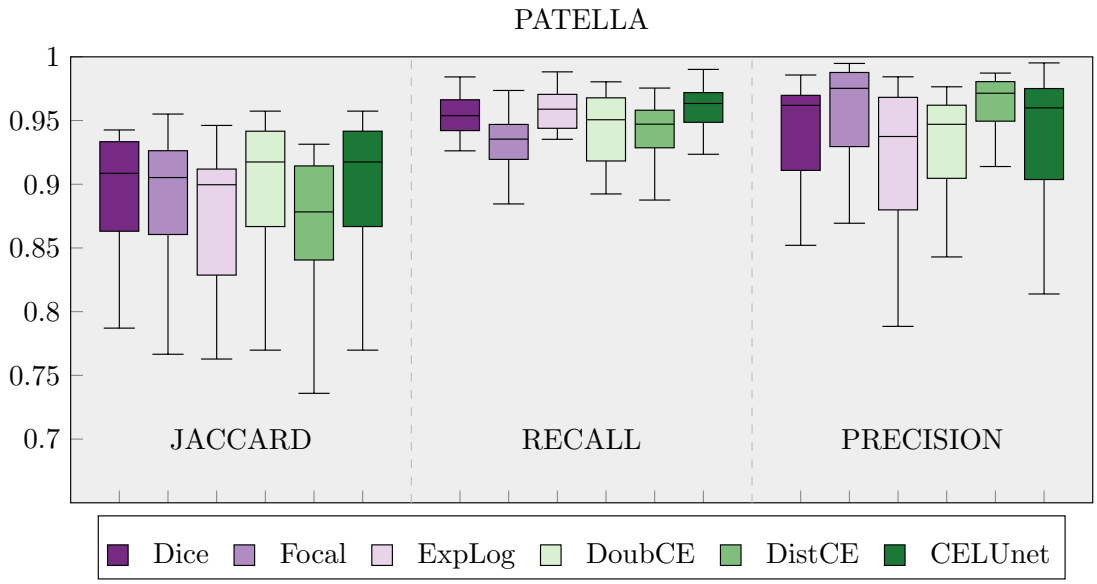


Figure 6.4. Boxplot of Jaccard, Recall and Precision score distributions for segmented patella.

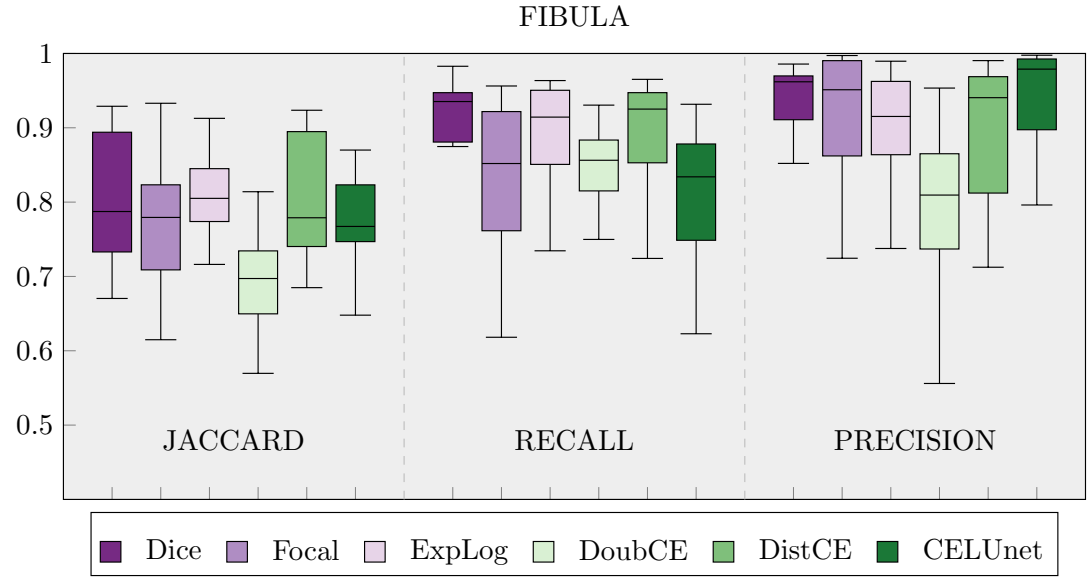


Figure 6.5. Boxplot of Jaccard, Recall and Precision score distributions for segmented fibula.

The statistical test was performed on Jaccard results of all 6 segmentation models considered, for both femur and tibia. The p-values produced by the Kruskal-Wallis tests are respectively of 9.9e-10 and 3.4e-9, denoting that the differences between some medians are statistically significant. The multiple comparison, performed afterwards with the tukey-kramer critical value, output a p-value for each possible 1 vs 1 comparison. For the femur, the most significant differences are found with CEL-Unet group against each one of Dice, Focal, ExpLog and DoubCE groups while no differences are found with respect to DistCE model. For the Tibia instead, the statistical difference provided by CEL-Unet results is weaker, since it is valid just against ExpLog and DoubCE. Tables 6.5 and 6.6 show results of the multiple comparison for femur and tibia, where bold numbers evidences results with statistical significance.

FEMUR: p-values

	Dice	Focal	ExpLog	DoubCE	DistCE	CEL-Unet
Dice	-	1.000	1.000	0.368	**0.043**	**3.9e-04**
Focal	-	-	1.000	0.356	**0.045**	**4.3e-04**
ExpLog	-	-	-	0.352	**0.046**	**4.4e-04**
DoubCE	-	-	-	-	**1.7e-05**	**3.2e-08**
DistCE	-	-	-	-	-	0.792
CEL-Unet	-	-	-	-	-	-

Table 6.5. Table reporting p-values for each possible comparison between femur results distributions.

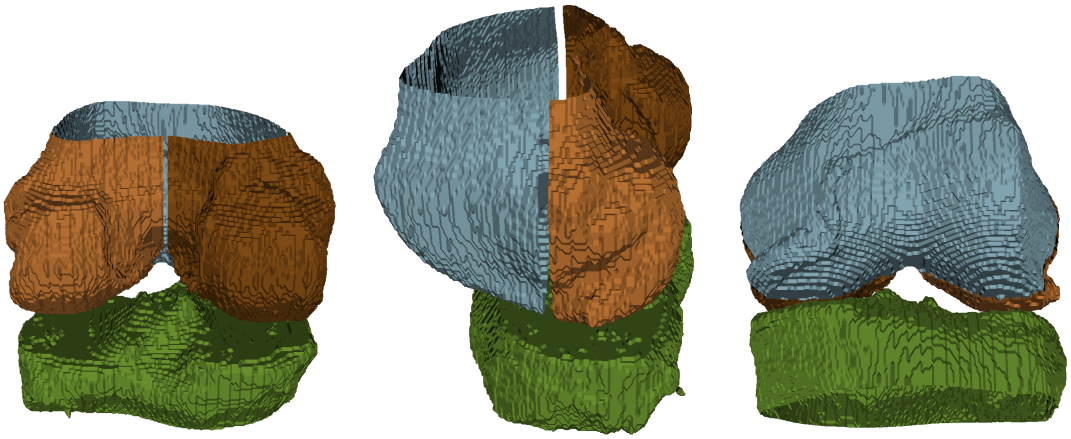
TIBIA: p-values

	Dice	Focal	ExpLog	DoubCE	DistCE	CEL-Unet
Dice	-	0.997	0.991	**9.8e-04**	0.788	0.139
Focal	-	-	0.897	**1.3e-04**	0.962	0.351
ExpLog	-	-	-	**0.009**	0.408	**0.028**
DoubCE	-	-	-	-	**2.0e-06**	**2.25e-08**
DistCE	-	-	-	-	-	0.858
CEL-Unet	-	-	-	-	-	-

Table 6.6. Table reporting p-values for each possible comparison between tibia results distributions.

6.3.2 Local Results

An initial inspection of the first segmentation results extracted by the loss functions comparison was made and it evidenced that volume matching between reconstructed and ground truth anatomies was very different depending on the location. As mentioned previously, the most critical areas are the ones closer to the knee joint, where the bones have been deformed and worn out by the long time rubbing. Moreover, these are the regions directly interested by Total Knee Artroplasty intervention. Hence, a local analysis was carried out in order to obtain segmentation results in 4 different delimited areas: right femur condyle, left femur condyle, femur trochlea and tibial plateau. These regions are shown in a posterior, lateral and frontal view in figure 6.6, each one marked in a different color. The goal was to see if the local analysis could provide any different result on the same reference metrics used for global segmentation assessment. For each one of these sub-regions, a table and a boxplot is reported, showing the mean of each metric across the test set and the approximate distribution.



(a) (b) (c)

Figure 6.6. Posterior (a), lateral (b) and frontal (c) views of the four regions defined for localized segmentation assessment.

Observing the results, it is possible to notice how the best scores of a local analysis match the best scores of the global one that corresponds to the same bone. This fact mainly evidences two aspects. Firstly, it tells that the segmentation errors found on the full anatomies are actually made almost only in these considered areas, which are very likely to be the main sources of mistake. Secondly, it proves that the best loss function and the CEL-Unet architecture, that achieve the best results, are able to improve segmentation performances precisely in these critical regions. This information is

RIGHT CONDYLE

	Jaccard	**Recall**	**Precision**
Dice	0.9364	0.9571	0.9777
Focal	0.9258	0.9330	**0.9919**
Exp Log	0.9403	0.9743	0.9644
Double CE	0.9078	**0.9843**	0.9214
Distanced CE	**0.9535**	0.9740	0.9785
CEL-Unet	**0.9601**	**0.9820**	0.9775

Table 6.7. Right condyle results.

LEFT CONDYLE

	Jaccard	**Recall**	**Precision**
Dice	0.9441	0.9663	0.9765
Focal	0.9432	0.9519	**0.9905**
Exp Log	0.9420	0.9780	0.9625
Double CE	0.9139	**0.9813**	0.9305
Distanced CE	**0.9573**	0.9792	0.9773
CEL-Unet	**0.9644**	**0.9857**	0.9783

Table 6.8. Left condyle results.

FEMUR TROCHLEA

	Jaccard	**Recall**	**Precision**
Dice	0.9383	0.9502	0.9869
Focal	0.9467	0.9556	**0.9901**
Exp Log	0.9446	**0.9832**	0.9599
Double CE	0.9305	0.9768	0.9514
Distanced CE	**0.9639**	0.9805	0.9829
CEL-Unet	**0.9713**	**0.9882**	0.9827

Table 6.9. Femur trochlea results.

crucially relevant as it differentiates the present condition from the case in which metric scores are enhanced thanks to a more accurate representation of less important areas or thanks to a sparse voxel adjustment, that would not provide any practical advantage in the end.

TIBIAL PLATEAU

	Jaccard	**Recall**	**Precision**
Dice	0.9330	0.9706	0.9603
Focal	0.9367	0.9556	0.9793
Exp Log	0.9334	0.9675	0.9637
Double CE	0.9108	**0.9742**	0.9334
Distanced CE	**0.9377**	0.9544	**0.9817**
CEL-Unet	**0.9554**	**0.9783**	0.9762

Table 6.10. Tibial plateau results.

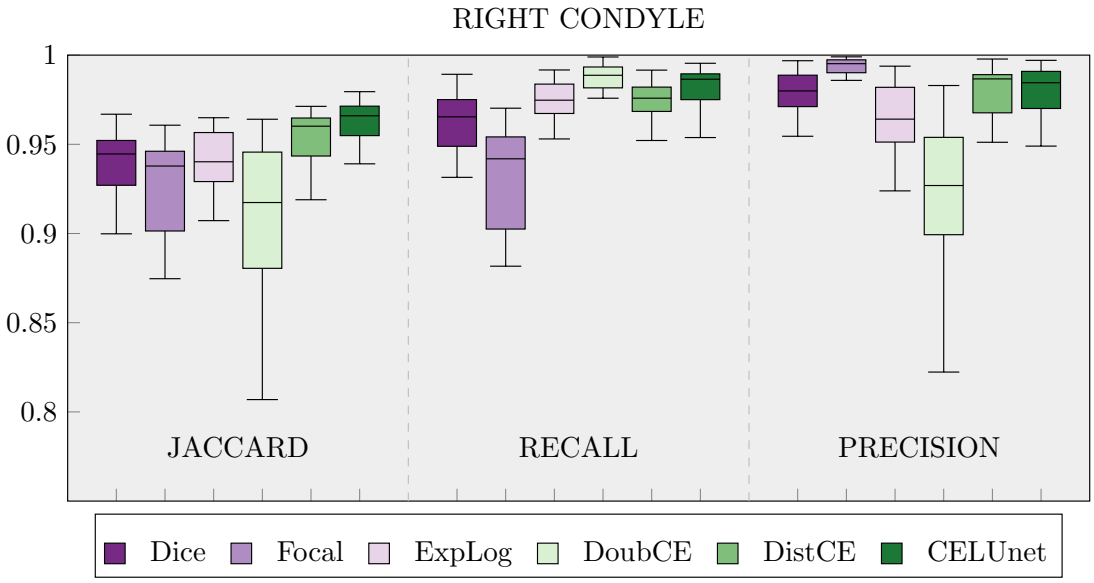


Figure 6.7. Boxplot of Jaccard, Recall and Precision score distributions for segmented right condyle.

The same statistical test mentioned at the end of section 6.3.1 was performed for each of the 4 separate regions, always on the Jaccard metric. Results were similar to the ones found in the global analysis regarding the same bone. As exposed before, the CEL-Unet test turns out to be the one that most frequently provides significant differences against median values of all other groups. This, again, is true for all femoral regions, but not for tibial plateau. Results of the test performed with CEL-Unet, as before, never find significant difference with Unet+DistCE results distribution.

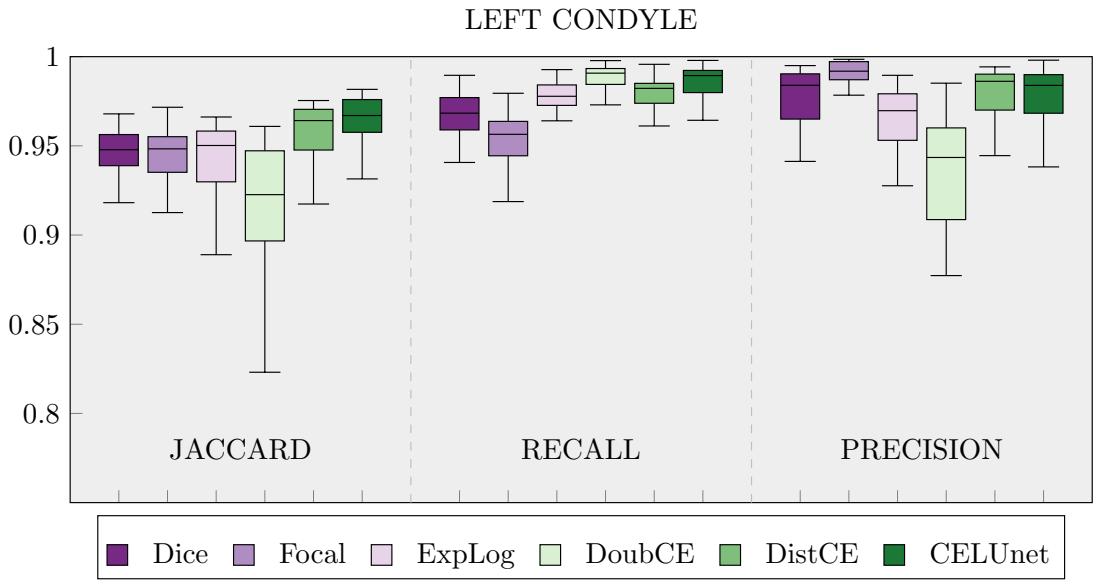


Figure 6.8. Boxplot of Jaccard, Recall and Precision score distributions for segmented left condyle.

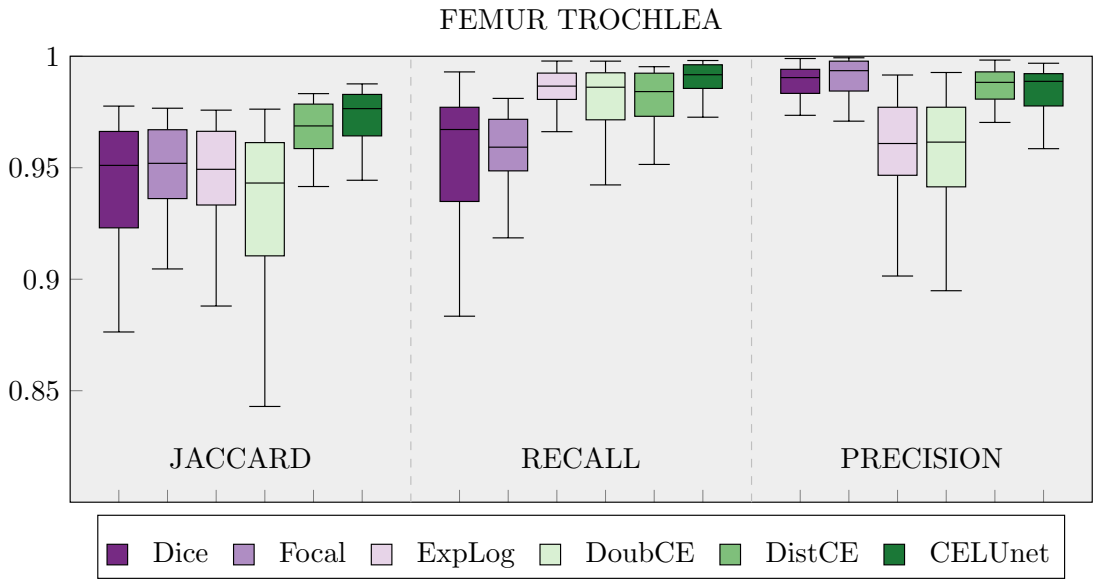


Figure 6.9. Boxplot of Jaccard, Recall and Precision score distributions for segmented femur trochlea.

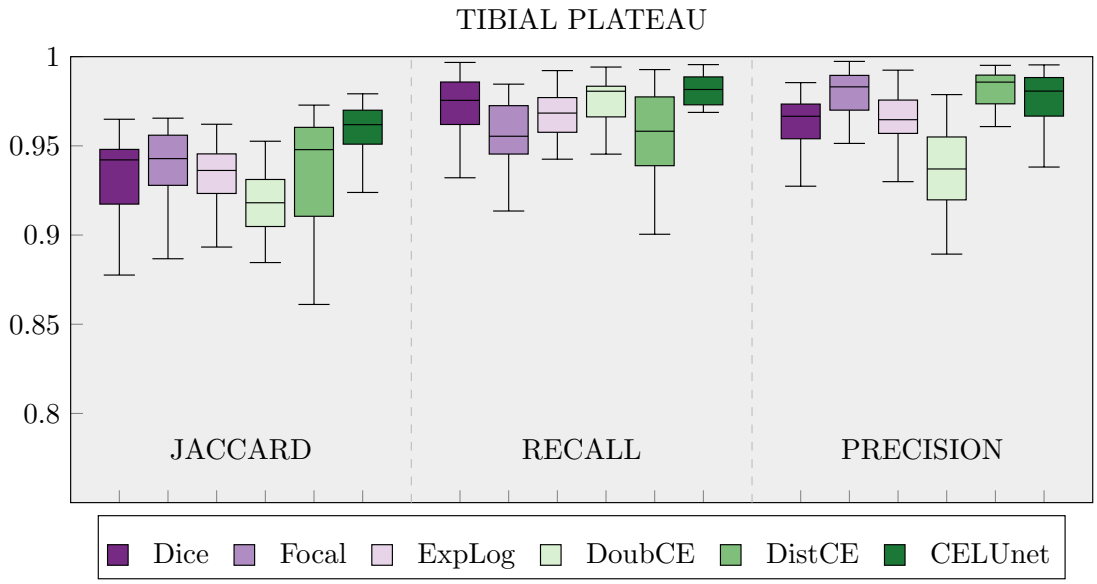


Figure 6.10. Boxplot of Jaccard, Recall and Precision score distributions for segmented tibial plateau.

6.3.3 Surface Analysis

In this section results of the distances between reconstructed and reference surfaces are reported. This is a key-analysis for the purposes of this work, as it provides further and more accurate information about the degree of similarity between the interested structures under comparison. Since the cutting jigs manufacturing process relies on the reconstructed surfaces of the bones, significant errors could bring to a mismatch between the guide and the real anatomy of the patient, which could then lead to a failure of the intervention. Maximum deviations from reference to reconstructed surfaces are computed with Hausdorff distance, while an average value is provided by the Root Mean Squared Error, for both global and local bone analyses. Results on femur and tibia are reported in the boxplots in figures 6.11 and 6.12, while results of localized surface analysis are presented in figures 6.13 and 6.14.

Outcomes of the surface evaluation of entire anatomies evidence how the CEL-Unet tends to minimize both Hausdorff and RMSE with respect to all other tests. This is achieved thanks to the strong focus on structures' boundaries that is given to such network during training with its double decoding path and the loss functions used. The performances of each of the 6 tests are similar on femur and tibia for both global and local analyses. There are slightly higher errors when tibial plateau

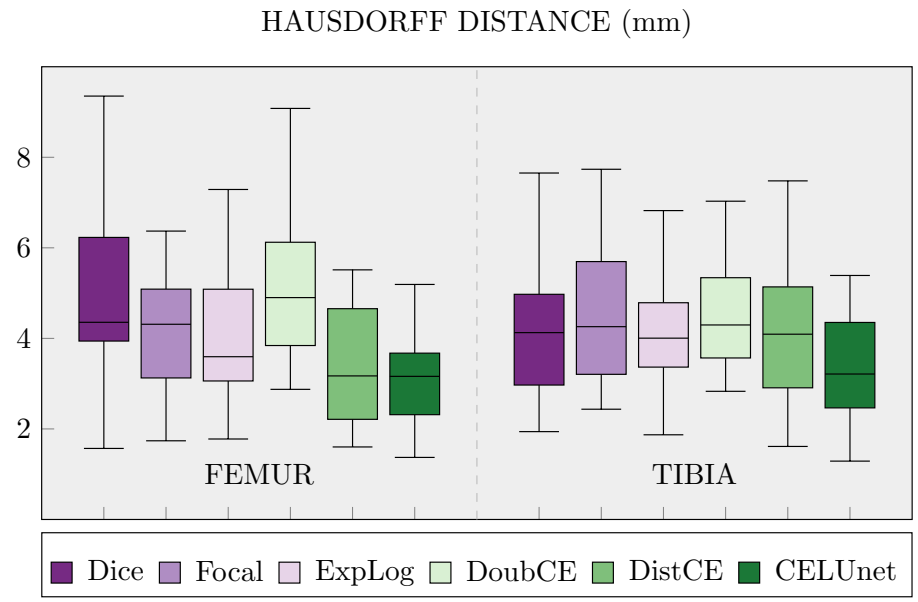


Figure 6.11. Boxplot of Hausdorff distance for tibia and femur.

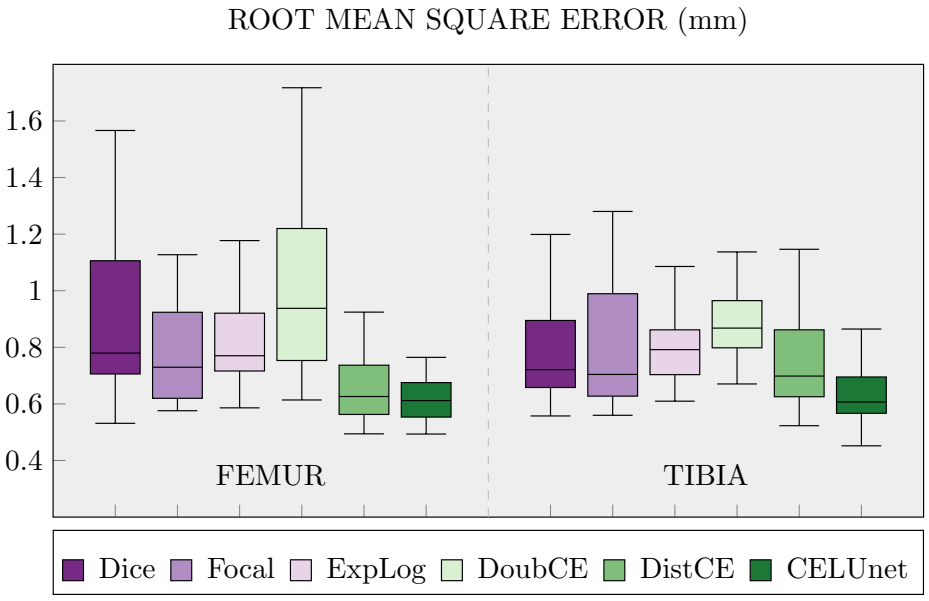


Figure 6.12. Boxplot of root mean square error of surface distance for tibia and femur.

is evaluated alone, which demonstrates the criticity of this area. However, again, CEL-Unet succeeds in decreasing these deviations with respect to other models.

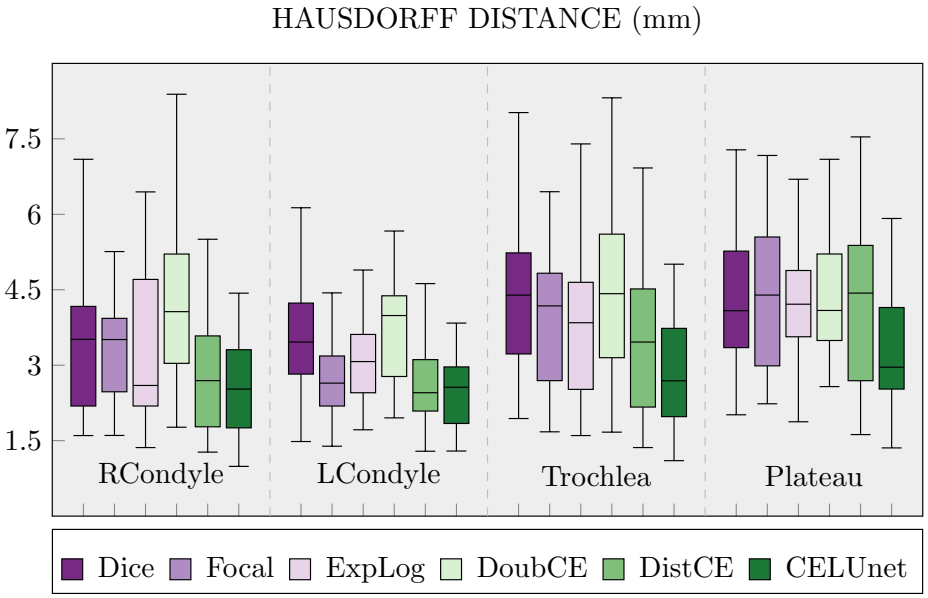


Figure 6.13. Boxplot of Hausdorff distance for left and right femur condyle, trochlea and tibial plateau.

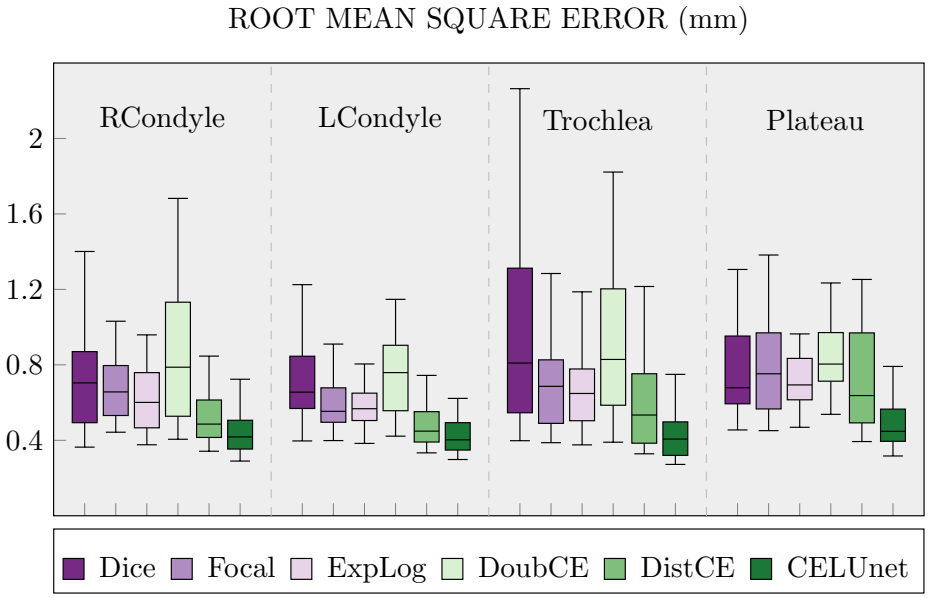


Figure 6.14. Boxplot of root mean square error of surface distance for left and right femur condyle, trochlea and tibial plateau.

6.4 Training timings

This section is dedicated to the exposition and comparison of training timings for the different models and loss functions. As a first consideration, it is worth to say that the choice of training 3D models strongly affected the timings, since it is much more expensive with respect to 2D model training. Anyway, 3D training boosted the performances significantly, hence the choice was mandatory, if satisfactory results were to be achieved. For what concerns Unet-based models, different loss functions provide different training times, depending on the expensiveness of the computations performed. Dice loss function was the fastest one, since it easily comprehends a first multiplication of the soft mask output for the ground truth volume and a summation of all voxel values then. Training with Focal, ExpLog and Double Cross Entropy losses took a few more hours, because they required some further computations to be done or some weight to be computed. Eventually, the Distanced Cross Entropy was surely the most slow and complicated to train. Indeed, to obtain the DWM, it is necessary to compute the Euclidean distance transform for each axial slice in the volume, which is quite onerous. Also, since the final loss balances the contributions of the Distanced Cross Entropy with the Dice loss, this last term still has to be computed and this fact further increases the time needed.
Regarding the training process of the CEL-Unet architecture, it surely was the most expensive and challenging one. As already mentioned in section 5.7, a workaround was used to fit all stored data in memory during training thanks to TensorFlow's mixed precision policy, which in some cases also helps to speed up computations, if specialized hardware for 16bits calculus are used. Anyway, the double decoding path of CEL-Unet contributed to increase the number of computations and hence also the training time. Furthermore, the two outputs produced by the network required a double supervision, and the use of two losses again affected the timings. Eventually, a great impact on the training time of the CEL-Unet was also given by the on-line generation of contour-ground truth labels for the output of the Edge decoding branch. This choice was forced by the impossibility of uploading the yet stored and previously computed contour labels during training, because of memory constraints. Expensive computation of the two losses for Mask and Edge output was also needed. Figure 6.15 shows the timings for each of the models, in terms of number of seconds needed to complete an epoch of training.

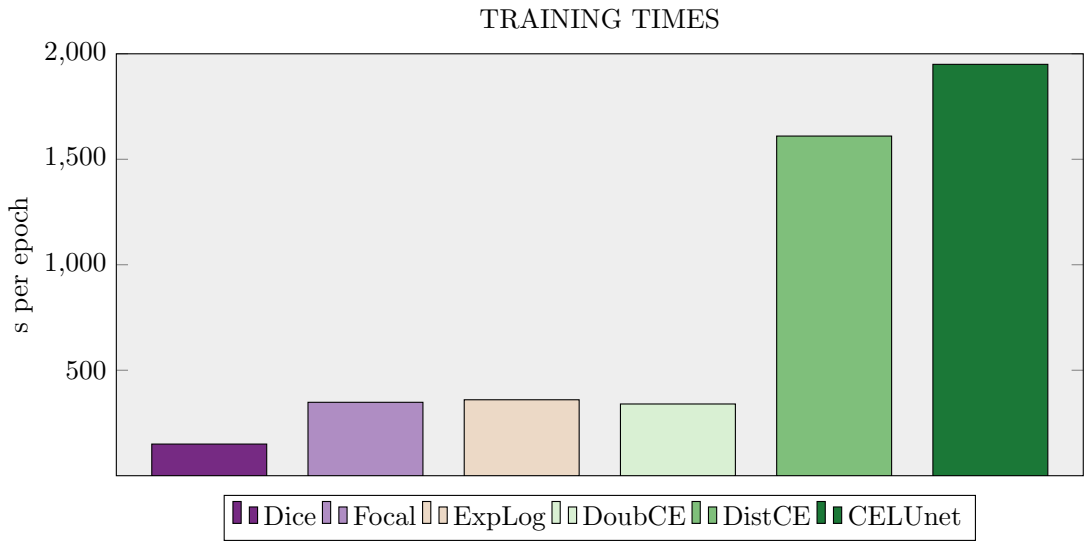


Figure 6.15. Comparison of training times for each model.

6.5 Back Analysis

After all models were trained and tested and all results were extracted to assess segmentation accuracy, a back analysis was carried out, with the aim of better understanding differences in models' performances. Since the test dataset contains a relatively small number of patient cases, it was possible to inspect volumes one by one and to extract some cases of analysis that can be of particular interest for our purposes and for future developments of either the present or similar work projects. Following this approach, the idea was to visually reconstruct the segmented surfaces obtained with each model, in order to compare them with each other and with the ground truth surface. The tool developed highlights over- and under-segmented regions by coloring each voxel according to its distance from the ground truth surface. This allows to understand how different models are able to segment the anatomies in critical locations that frequently correspond to the narrow intra-articular space, the area around the tibial plateau or the area around the femur condyle. Fibula and patella bones were excluded in this visual analysis in order to enhance the focus on femur and tibia anatomies. Also, for visual reasons, one of the models was excluded. It was chosen to exclude the one that provides the worst performance, which is the Unet model trained with the Double Cross Entropy Loss.
In particular, two cases among the ones in the test set were carefully chosen and are reported in this section: case patient 387 and case patient 391. Since it was

infeasible to show all of them in this document, the criterion of the choice was to show the examples that can provide evidences about how different models produce different segmentation results. The case 387 represents one of the most challenging: in figures 6.16 and 6.17, it is possible to notice how the shapes of tibia and femur are strongly irregular and far from the physiological ones. Also, the intra-articular space is almost absent, and this causes femur and tibia bones to come into contact, which renders the segmentation even more complicated. Case patient 391, instead, shows a more regular global anatomy of the two bones, but it presents a visible round-shaped osteophyte, localized in the area between the medial and the lateral condyle. Hence, it is interesting to see how the models consider this structure differently and to see which one can provide the most realistic reconstruction. Again, posterior and frontal views of ground truth and reconstructed surfaces of patient 391 are shown in figure 6.18 and figure 6.19.

Every voxel of the surfaces represented in the mentioned figures is colored according to its distance from the target ground truth surface. A voxel is assigned a negative distance value when under segmentation occurs. In this case, it takes a color in the blue scale, where brighter blues correspond to a shorter distance, and darker blues correspond to a longer one. On the contrary, the distance value is positive when the anatomy is over-segmented, and voxel colors are picked, in this case, in the red scale, with the same policy. The darkest red and blue colors correspond respectively to positive and negative errors of 4 millimeters. Even if greater errors than 4 millimeters are found in the segmentations, this value was chosen because it provides the most comprehensive representation, allowing also the visualization of slightly over and under-segmented voxels.

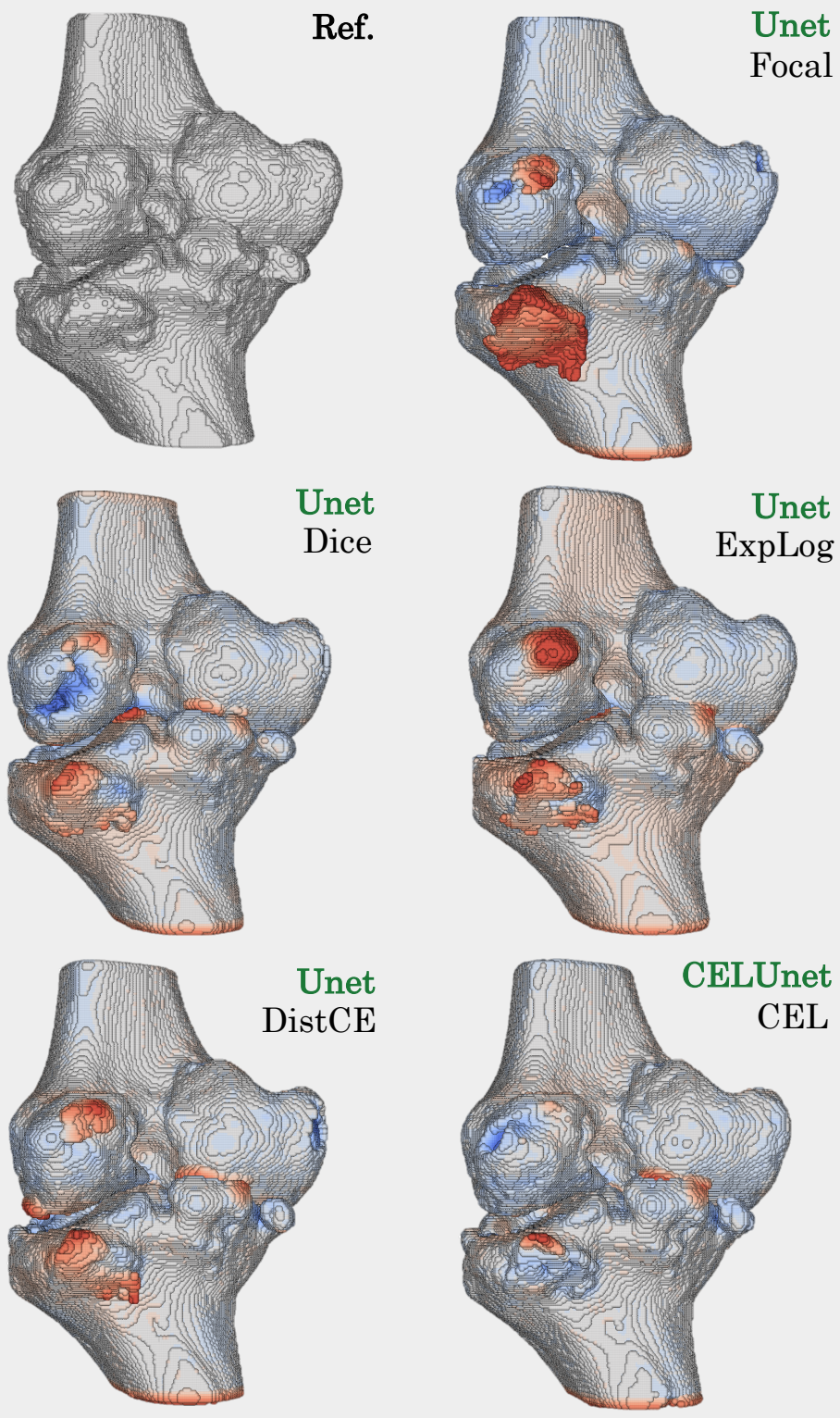


Figure 6.16. Case Patient 387, posterior view. Top Left: reference surface. Top right: Unet trained with Focal Loss. Middle Left: Unet trained with Dice Loss. Middle Right: Unet trained with ExpLog loss. Bottom Left: Unet trained with DistCE loss. Bottom Right: CEL-Unet trained with Combined Edge Loss.

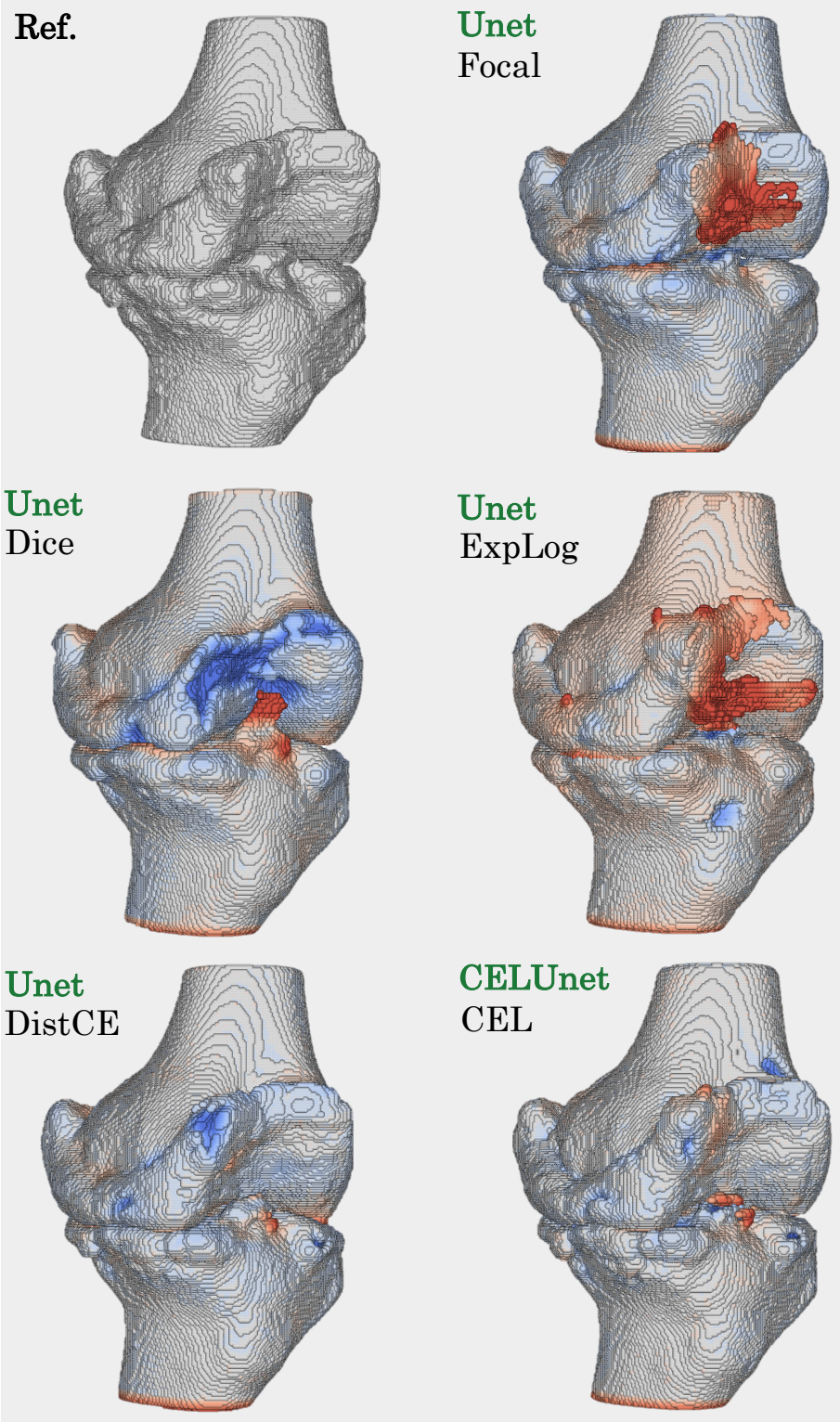


Figure 6.17. Case Patient 387, frontal view. Top Left: reference surface. Top right: Unet trained with Focal Loss. Middle Left: Unet trained with Dice Loss. Middle Right: Unet trained with ExpLog loss. Bottom Left: Unet trained with DistCE loss. Bottom Right: CEL-Unet trained with Combined Edge Loss.

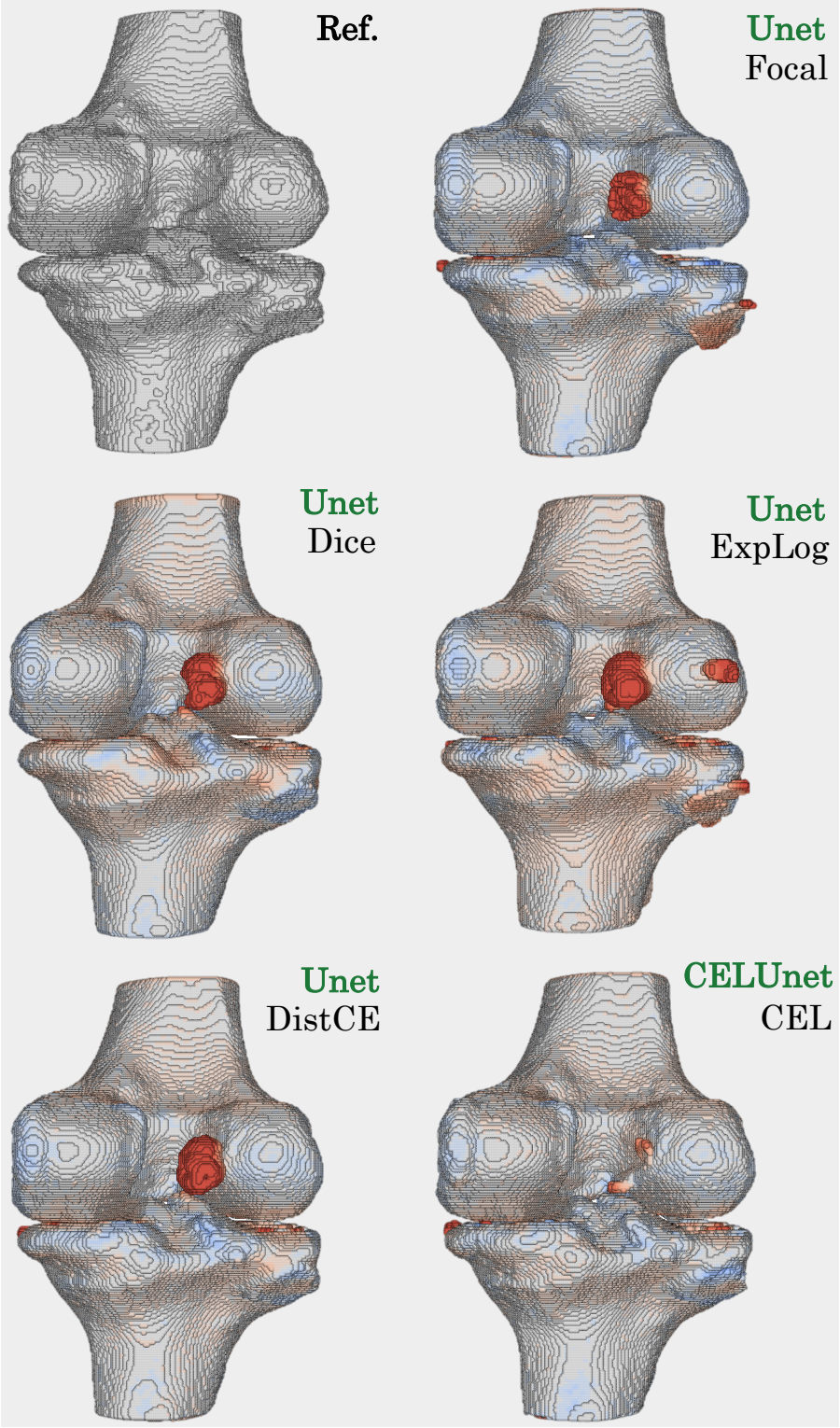


Figure 6.18. Case Patient 391, posterior view. Top Left: reference surface. Top right: Unet trained with Focal Loss. Middle Left: Unet trained with Dice Loss. Middle Right: Unet trained with ExpLog loss. Bottom Left: Unet trained with DistCE loss. Bottom Right: CEL-Unet trained with Combined Edge Loss.

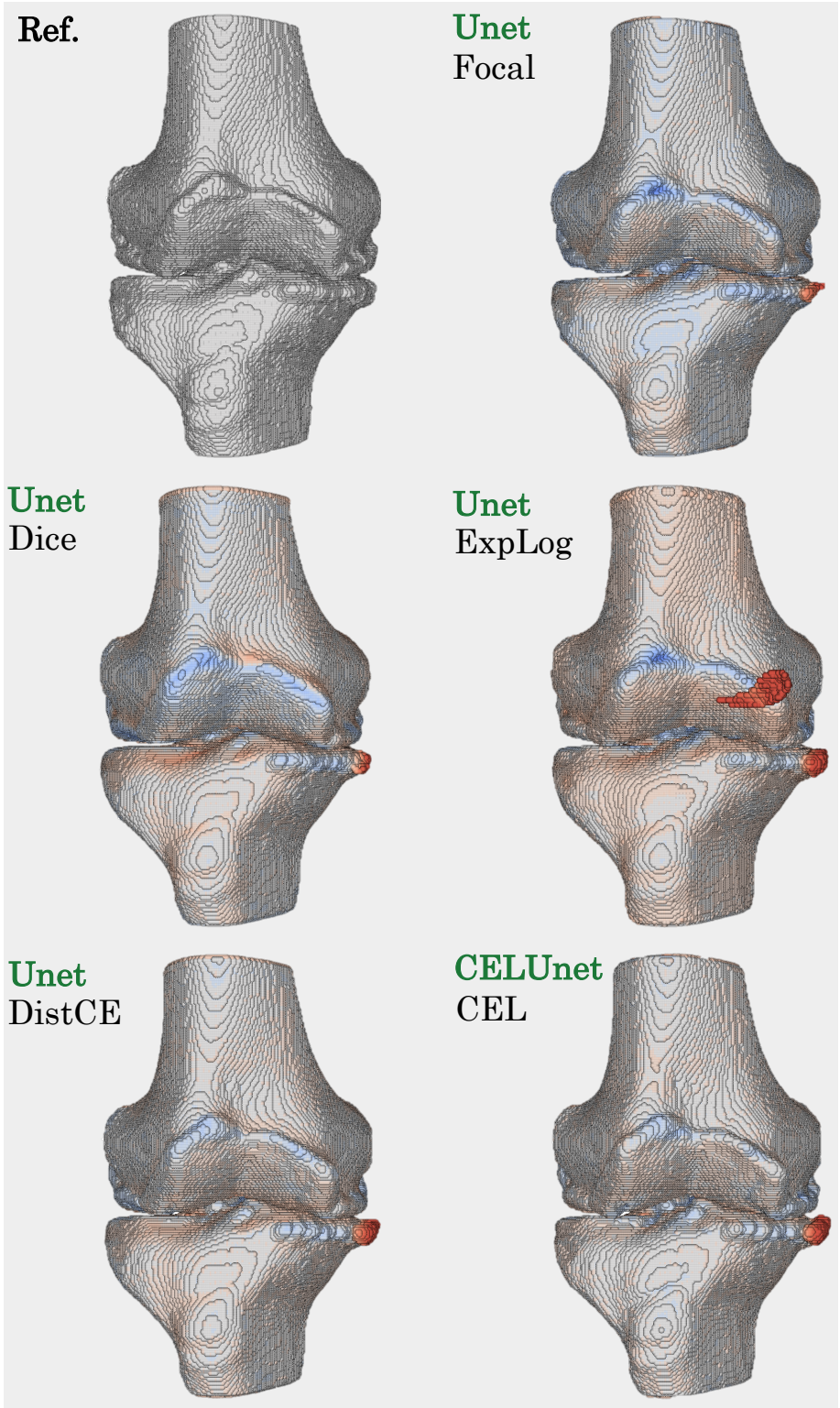


Figure 6.19. Case Patient 391, frontal view. Top Left: reference surface. Top right: Unet trained with Focal Loss. Middle Left: Unet trained with Dice Loss. Middle Right: Unet trained with ExpLog loss. Bottom Left: Unet trained with DistCE loss. Bottom Right: CEL-Unet trained with Combined Edge Loss.

To summarize these results in one sentence: the less the color, the better. With a general overview, this tool can confirm and enforce results reported in the tables with the target metrics. For both patient cases, the blueish color obtained in the surface reconstructed with the Focal loss, evidences the tendency of this cost function to under-segment anatomies. In the same way, the reddish color of the surface corresponding to ExpLog loss suggests over-segmentation is generally present.
Focusing on case 387, this visualization allows to see that the segmentation in correspondence of the tibio-fibular joint is often a bit rough and inaccurate. This is caused by the fact that the space between the two bones has almost disappeared, as it happens for the intra-articular space. For this reason, segmentation models sometimes fail in finding the correct separation between the two anatomies. The same error happens in correspondence of the patello-femoral groove, where part of the patella is considered as femur and this fact produces over-segmentation voxels marked in red, as it is visible in figure 6.17. On the other hand, in this case the Dice loss happens to be way more conservative, by correctly excluding the patella from femur segmentation. Nevertheless, it strongly under-segment femur trochlea, as represented by the visible presence of blue voxels in that area. Additional misclassified voxels then are found around femur condyle area.
For case patient 391, the main focus is given to the ostophyte located in the internal area of the lateral femur condyle. This structure is not a part of any of the knee bones, as it can be noted in the reference surface, but most of the models are fooled by its appearance and tend to include it as part of the femur. On the other hand, it is relevant to notice that the CEL-Unet model is able to recognize it as extraneous and thus to achieve a more accurate result. In the frontal view instead, there are no relevant errors to consider, as all models manage to produce a realistic reconstruction of the bones.
A two dimensional comparison is also reported, which regards two significant axial sections of the CT volume of case patient 391, one in correspondence of the femur and one of the tibia. It is represented in figure 6.20 and figure 6.21 and it has the aim to visualize how CEL-Unet outperforms the other trained models in correspondence of these critical axial coordinates. In both figures, the yellow circles evidence the spots where CEL-Unet model manage to improve segmentation accuracy, by identifying small and tricky details that are missed by other analyses. In figure 6.20, these spots regard the previously mentioned osteophyte and a corner of the patella. In figure 6.21 instead, the axial slices represent the terminal part on the top of the tibial plateau, where the anatomy becomes strongly uneven.

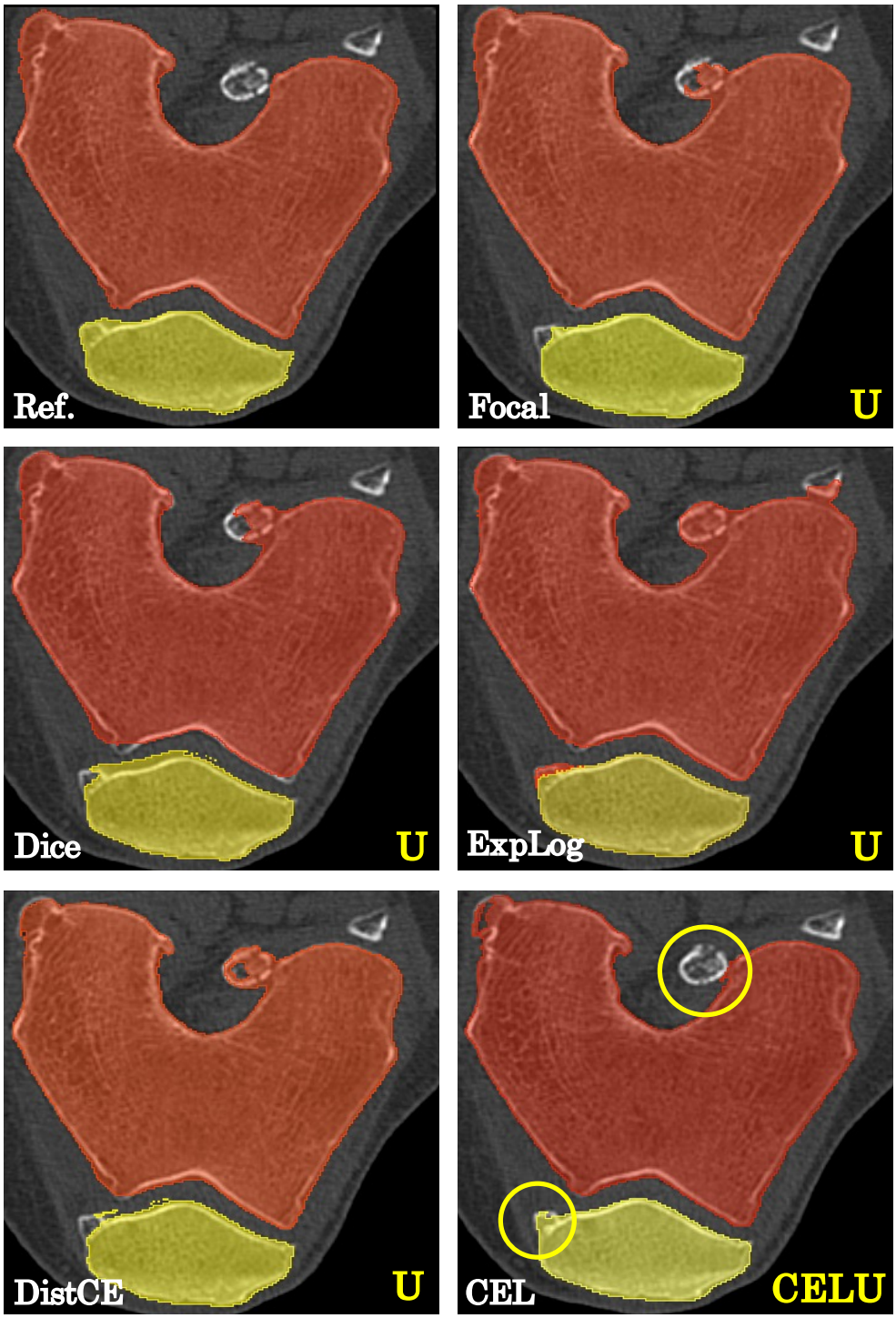


Figure 6.20. Case Patient 391, 2D slice showing femur and patella. Top Left: reference segmentation. Top right: Unet trained with Focal Loss. Middle Left: Unet trained with Dice Loss. Middle Right: Unet trained with ExpLog loss. Bottom Left: Unet trained with DistCE loss. Bottom Right: CEL-Unet trained with Combined Edge Loss.

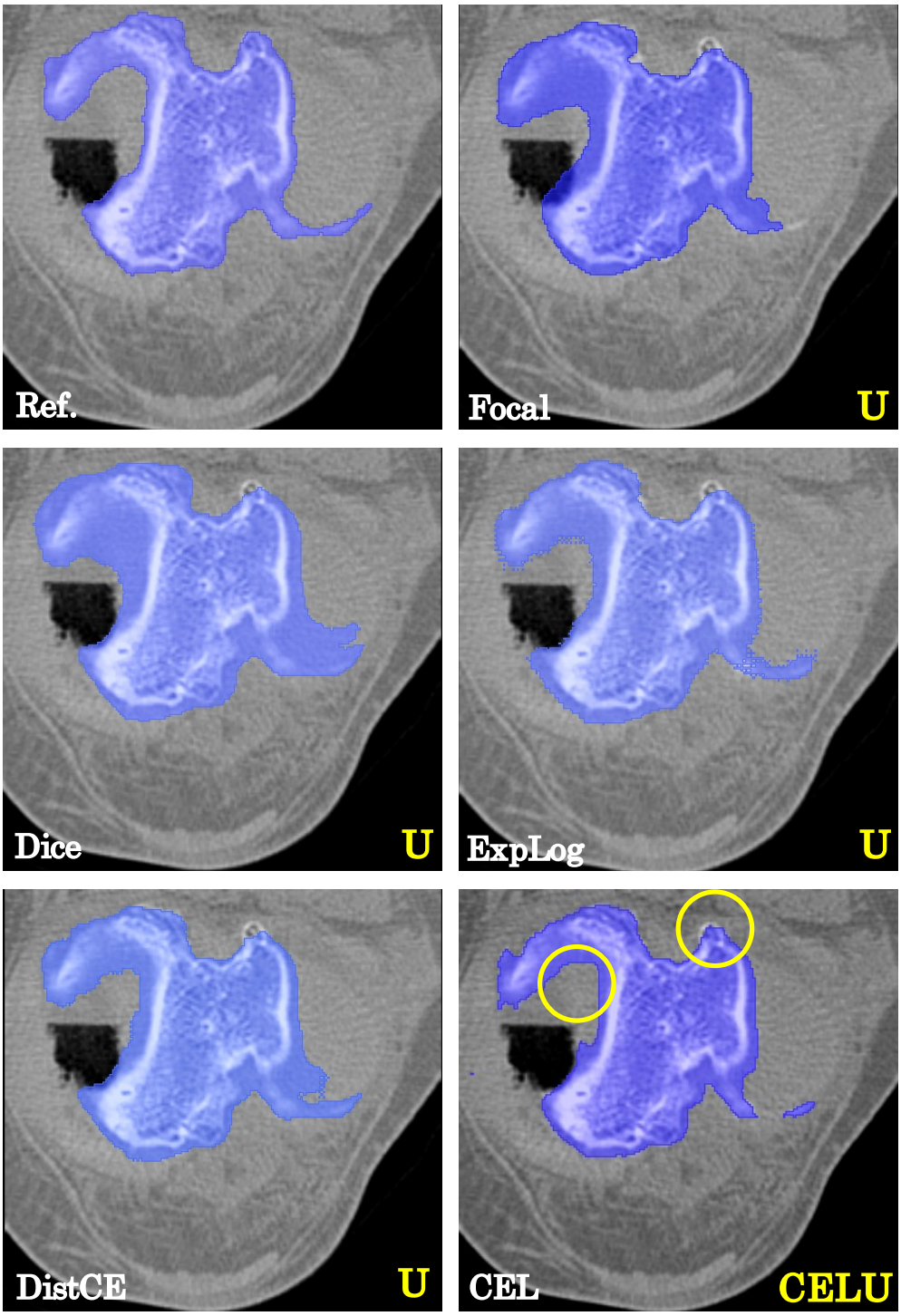


Figure 6.21. Case Patient 391, 2D slice showing tibial plateau. Top Left: reference segmentation. Top right: Unet trained with Focal Loss. Middle Left: Unet trained with Dice Loss. Middle Right: Unet trained with ExpLog loss. Bottom Left: Unet trained with DistCE loss. Bottom Right: CEL-Unet trained with Combined Edge Loss.

Chapter 7

Discussion

7.1 Main Findings

The present work was developed with the aim of inspecting the possibility of enhancing segmentation results provided by classical state of the art deep learning model such as the Unet. The approach chosen was to firstly find the most suitable loss function for model training, and then to transfer this objective onto a new tailored architecture to further boost accuracy of the results. Among all possible directions of research in deep learning field, considered the clinical context of this work, the strong boundary attention was taken as guideline and it brought to satisfactory results. For all bones included in the analysis, the new method outperformed all other approaches and succeeded in refining segmentation in the most critical areas. Localized evaluation also clearly shows the improvements achieved with CEL-Unet architecture. This fact can be taken as a proof that the development of current deep learning models has not yet reached the bottleneck and that every punctual application can be tackled with the most suitable solution to take results to the best.

CEL-Unet architecture arises from a very simple idea, and it is realized in a straightforward way, which does not add any conceptual complexity to the model, as it sometimes happens in the recent literature. This consideration gains importance in the biomedical field, where the development of a deep learning-based marketable solution needs to be evaluated and deeply discussed with physicians, who usually have no expertise or technical competences on the topic. Simple, effective and understandable solutions are in this case well-seen and usually preferred with respect to the more

articulated ones. In our case also, interpretability of the model is gained, since it produces a double output, that can be inspected and can provide additional clues about hidden computations that lead to the final segmentation.
CEL-Unet was the only model trained with 3 depth levels, while all others Unet models trained with the different loss functions were built with 5 levels of depth. Despite this, CEL-Unet was able to outperform all others. This fact leaves a margin for further boost the performances by trying to increase the depth, in the case greater computational resources are available.

Patella and fibula were included in the segmentation of the knee joint anatomy, more for scientific reasons than for clinical ones. As already mentioned actually, patella and fibula do not take part in preoperative planning of TKR intervention, so their reconstruction is less than needed. However, the choice was made to exploit at best the available dataset, that also included label files for these anatomies, and to test CEL-Unet on a more challenging task, with respect to the segmentation of just the tibia and the femur. A qualitative analysis was performed and a few quick trials (not presented here) were carried out which demonstrated that accuracy achieved on tibia and femur is slightly increased, with the same architecture, when the patella and the fibula are excluded from the segmentation. This can probably be explained by the fact that the inferior number of classes reduces the amount of information to analyze. Hence, given the fact that our network is neither very wide nor very deep, this allows the hyperparameters to build a better representation of input data and to achieve better results.

7.2 Comparison with the Literature

Many researches already demonstrated the feasibility of automated bone segmentation using deep learning and neural networks. In [48] authors present a CNN for segmentation of the spine in CT scans, with a provided sensitivity of 97% and 3D surface distance error of 7.4mm. The study was performed on a small dataset of 32 patients that strongly reduced the extent of the results. A Dice similarity coefficient of about 97% regarding femur segmentation was recently reported in [31], with a dataset of 150 patients. Differently from the present one, the work performed a binary segmentation that just considers one single bone and used a dataset with a reduced

inter-slice resolution of about 3mm. This can raise some doubts about the quality of the 3D reconstruction attainable on the basis of such a segmentation. The Unet architecture was also used in [32] for bone segmentation of 53 low-quality, whole-body CT scans, and provided a dice score of 95%. However, the dataset was acquired with a unique scanner, so the generalization of these results can be considered reduced. In our work, we conversely used volumes acquired with four different scanners, namely Philips, Canon Medical Systems, GE Medical Systems and Toshiba. Another specific study used a lightweight Unet architecture for segmentation of pediatric hands in X-ray images, reporting 94% of sensitivity.
Overall, it is possible to say that results of this thesis project are in line with the presented literature. Besides, the mentioned researches in many cases apply for a wider clinical context and usually do not show the same degree of pathological severity found in the present work, which strongly contributes to make the task even more challenging.

7.3 Technical Challenges

3D volume segmentation requires a huge amount of computational power and takes a very long time to be accomplished. The shortest training process took six hours and fifteen minutes, while the longest, which regards the CEL-Unet, took up to 65 hours. Our resources relied on Google Colab platform, which is a great solution that provides free GPU usage, but in a limited amount. For this reason, the mentioned training times are actually lower than the real ones, which also comprehend all the forced interruptions of the process, that had to be manually recovered. On Google Colab indeed, continuous training can proceed for a maximum of 12 hours, after which the user is disconnected and all stored data and allocated memory are lost. Progress of all models was continuously saved during training and the latest was recovered after an interruption occurred, in order to complete the procedure. However, this strongly contributed to increase the time needed to reach the final number of epochs. Hence, it was possible to evaluate the models only after some hours or some days of training, which led to the difficulty of performing fast hyperparameter tuning and limited the possibility of trying different configurations for each model.

Memory constraints were also very strict, mainly in the second part of this work, where the CEL-Unet was developed and ready to be trained. The double decoding

path increased memory requirements and made it necessary to use the mixed precision TensorFlow policy, to store tensors in 16bits, when possible. The depth of the network was decreased to 3 and the number of initial filters was maintained equal to 8, value that influences the subsequent number of feature maps produced at each stage in the network. This number is significantly lower than what is usually found in the literature, but it also depends on the number of classes of the segmentation case. In this work, it was decided to stick to this value without trying to decrease it, in order to maintain a value which was higher than the number of classes (5) by a few units. However, given that patella and fibula were found irrelevant for the present clinical purposes, a segmentation including just 3 classes could be set, with a deeper network and a lower number of initial filters. These configurations were not considered in this work, due to time constraints caused by the extremely long training of CEL-Unet. Maximum batch size was equal to 2, no bigger batches could be set without running into memory overflow.

Chapter 8

Conclusions

In this thesis, a successful novel deep learning architecture is presented for segmentation of the distal femur and the proximal tibia in patients with severe osteoarthritis, from CT volumes acquired throughout different scanners. The task is challenging due to the pathological conditions of the bones and to the high level of accuracy needed to bring such automated solutions closer to the market. Together with the great amount of research that is going on in this field, this work wants to underline the strength and the effectiveness that deep learning-based systems can have as support tools for clinical decision making, pre-operative planning and diagnosing. Results obtained were shown valuable in terms of segmentation and surface reconstruction, being comparable to results achievable by means of expert segmentation. The effort to train dependable models can be great, but the advantages are surely huge.
Therefore, it is possible to argue that leveraging deep learning architectures in clinical tools can help in reducing sensibly time and efforts for medical image segmentation, still providing high accuracy and great reliability for performing pre-operative planning of interventions such as PSI-based Total knee Arthroscopy.

8.1 Future Developments

The limitations of this work of thesis, mainly encountered due to time constraints, leave much space for further developments of the present research. As mentioned, the impossibility to perform a high number of trials on the proposed architecture would suggest that a finer hyperparameter tuning could potentially push forward the

segmentation performances even more. Besides, by increasing the numerosity of the training and testing data more robust models and more reliable evaluations could be reached.
Moreover, it would be interesting to test the proposed architecture on different anatomies to see if satisfactory results can still be achieved. At the same time, the CEL-Unet could be extended to the segmentation of MR images, that provide much greater details on soft tissues with respect to CT scans.
In a broader perspective also, this segmentation model could be exploited to build a tool for automatic segmentation, in order to support radiologists and to substitute the tedious manual delineation or the use of expensive softwares. The rapidity with which a deep learning model can produce segmentation outputs would speed up the process and would just require a final editing of the segmentation, by the hand of an expert operator. In a scenario in which the segmentation model could be deployed into a marketable product, it would be of great interest and utility to direct the research towards the so-called Interpretable Artificial Intelligence. Such solutions can provide explanations, at a certain level, regarding how the model's output is computed and can therefore facilitate humans to trust information provided by these intelligent algorithms.

Acronyms

ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
OA	Osteoarthritis
TKA	Total Knee Arthroplasty
PSI	Personalized Surgical Instrumentation
TKR	Total Knee Replacement
PMT	Patient Matched Technology
MLP	Multi Layer Perceptron
CE	Cross Entropy
DSC	Dice Similarity Coefficient
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
FCN	Fully Convolutional Network
EDT	Euclidean Distance Transform
DWM	Distance Weight Map
CEL	Combined Edge Loss

Bibliography

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28 (cit. on pp. 2, 28).

[2] Fabian Isensee et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature Methods* 18.2 (Dec. 2020), pp. 203–211. DOI: 10.1038/s41592-020-01008-z (cit. on pp. 2, 31).

[3] Gloria Phillips-Wren and Lakhmi Jain. "Artificial Intelligence for Decision Making". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 531–536. DOI: 10.1007/11893004_69 (cit. on p. 2).

[4] Antoine Buetti-Dinh et al. "Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition". In: *Biotechnology Reports* 22 (June 2019), e00321. DOI: 10.1016/j.btre.2019.e00321 (cit. on p. 3).

[5] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Dec. 2015. DOI: 10.1109/iccv.2015.123 (cit. on p. 3).

[6] Matthew A Varacallo, T.D.Luo, and N. Johanson. "Total Knee Arthroplasty (TKA) Techniques". In: 2019 (cit. on p. 7).

[7] Neil P. Sheth, Adeel Husain, and Charles Lenwood Nelson. "Surgical Techniques for Total Knee Arthroplasty". In: *Journal of the American Academy of Orthopaedic Surgeons* 25.7 (July 2017), pp. 499–508. DOI: 10.5435/jaaos-d-14-00320 (cit. on p. 7).

[8] B. Boonen et al. “No difference in clinical outcome between patient-matched positioning guides and conventional instrumented total knee arthroplasty two years post-operatively”. In: *The Bone & Joint Journal* 98-B.7 (July 2016), pp. 939–944. DOI: 10.1302/0301-620x.98b7.37274 (cit. on p. 8).

[9] Kiriakos Daniilidis and Carsten O. Tibesku. “A comparison of conventional and patient-specific instruments in total knee arthroplasty”. In: *International Orthopaedics* 38.3 (July 2013), pp. 503–508. DOI: 10.1007/s00264-013-2028-9 (cit. on p. 8).

[10] Lorenzo Mattei et al. “Patient specific instrumentation in total knee arthroplasty: a state of the art”. In: *Annals of Translational Medicine* 4.7 (Apr. 2016), pp. 126–126. DOI: 10.21037/atm.2016.03.33 (cit. on p. 8).

[11] Werner Anderl et al. “Patient-specific instrumentation improved mechanical alignment, while early clinical outcome was comparable to conventional instrumentation in TKA”. In: *Knee Surgery, Sports Traumatology, Arthroscopy* 24.1 (Oct. 2014), pp. 102–111. DOI: 10.1007/s00167-014-3345-2 (cit. on p. 10).

[12] M. L. Dao Trong et al. “Improved positioning of the tibial component in unicompartmental knee arthroplasty with patient-specific cutting blocks”. In: *Knee Surgery, Sports Traumatology, Arthroscopy* 23.7 (Jan. 2014), pp. 1993–1998. DOI: 10.1007/s00167-014-2839-2 (cit. on p. 10).

[13] A.O. Erdogan, N.S. Gokay, and A. Gokce. “Preoperative Planning of Total Knee Replacement”. In: *Arthroplasty - Update*. InTech, Feb. 2013. DOI: 10.5772/55023 (cit. on p. 10).

[14] Jasvinder A. Singh et al. “Rates of Total Joint Replacement in the United States: Future Projections to 2020–2040 Using the National Inpatient Sample”. In: *The Journal of Rheumatology* (2019). ISSN: 0315-162X. DOI: 10.3899/jrheum.170990. eprint: https://www.jrheum.org/content/early/2019/04/09/jrheum.170990.full.pdf. URL: https://www.jrheum.org/content/early/2019/04/09/jrheum.170990 (cit. on pp. 10, 11).

[15] M.C.S. Inacio et al. “Projected increase in total knee arthroplasty in the United States – an alternative projection model”. In: *Osteoarthritis and Cartilage* 25.11 (Nov. 2017), pp. 1797–1803. DOI: 10.1016/j.joca.2017.07.022 (cit. on p. 10).

[16] Takeshi Ogawa et al. “Factors related to disagreement in implant size between preoperative CT-based planning and the actual implants used intraoperatively for total hip arthroplasty”. In: *International Journal of Computer Assisted Radiology*

and Surgery 13.4 (Dec. 2017), pp. 551–562. DOI: 10.1007/s11548-017-1693-3 (cit. on p. 12).

[17] Brahim Ait Skourt, Abdelhamid El Hassani, and Aicha Majda. "Lung CT Image Segmentation Using Deep Neural Networks". In: *Procedia Computer Science* 127 (2018), pp. 109–113. DOI: 10.1016/j.procs.2018.01.104 (cit. on p. 19).

[18] Adam Paszke et al. "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation". In: *Arxiv* (June 2016) (cit. on p. 19).

[19] Hidenori Ide and Takio Kurita. "Improvement of learning for CNN with ReLU activation by sparse regularization". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, May 2017. DOI: 10.1109/ijcnn.2017.7966185 (cit. on p. 20).

[20] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (Jan. 1979), pp. 62–66. DOI: 10.1109/tsmc.1979.4310076 (cit. on p. 24).

[21] Yuncong Feng et al. "A multi-scale 3D Otsu thresholding algorithm for medical image segmentation". In: *Digital Signal Processing* 60 (Jan. 2017), pp. 186–199. DOI: 10.1016/j.dsp.2016.08.003 (cit. on p. 24).

[22] Hima Bindu. "An improved medical image segmentation algorithm using OTSU Method". In: *SHORT PAPER International Journal of Recent Trends in Engineering* 2 (Dec. 2009) (cit. on p. 24).

[23] Hima Bindu and K. Prasad. "An Efficient Medical Image Segmentation Using Conventional OTSU Method". In: *International Journal of Advanced Science and Technology* 38 (Jan. 2012) (cit. on p. 24).

[24] Boykov and Jolly. "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. IEEE Comput. Soc. DOI: 10.1109/iccv.2001.937505 (cit. on p. 26).

[25] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models". In: *International Journal of Computer Vision* 1.4 (Jan. 1988), pp. 321–331. DOI: 10.1007/bf00133570 (cit. on p. 26).

[26] E. Mortensen et al. "Adaptive boundary detection using 'live-wire' two-dimensional dynamic programming". In: *Proceedings Computers in Cardiology*. IEEE Comput. Soc. Press. DOI: 10.1109/cic.1992.269378 (cit. on p. 26).

[27] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. "Generative Image Segmentation Using Random Walks with Restart". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 264–275. DOI: 10.1007/978-3-540-88690-7_20 (cit. on p. 27).

[28] R. Adams and L. Bischof. "Seeded region growing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.6 (June 1994), pp. 641–647. DOI: 10.1109/34.295913 (cit. on p. 27).

[29] Lian Ding et al. "A Lightweight U-Net Architecture Multi-Scale Convolutional Network for Pediatric Hand Bone Segmentation in X-Ray Image". In: *IEEE Access* 7 (2019), pp. 68436–68445. DOI: 10.1109/access.2019.2918205 (cit. on p. 30).

[30] Bingjiang Qiu et al. "Automatic segmentation of the mandible from computed tomography scans for 3D virtual surgical planning using the convolutional neural network". In: *Physics in Medicine & Biology* 64.17 (Sept. 2019), p. 175020. DOI: 10.1088/1361-6560/ab2c95 (cit. on p. 30).

[31] Fang Chen et al. "Three-Dimensional Feature-Enhanced Network for Automatic Femur Segmentation". In: *IEEE Journal of Biomedical and Health Informatics* 23.1 (Jan. 2019), pp. 243–252. DOI: 10.1109/jbhi.2017.2785389 (cit. on pp. 30, 84).

[32] André Klein et al. "Automatic bone segmentation in whole-body CT images". In: *International Journal of Computer Assisted Radiology and Surgery* 14.1 (Nov. 2018), pp. 21–29. DOI: 10.1007/s11548-018-1883-7 (cit. on pp. 30, 85).

[33] Yilong Chen et al. "Channel-Unet: A Spatial Channel-Wise Convolutional Neural Network for Liver and Tumors Segmentation". In: *Frontiers in Genetics* 10 (Nov. 2019). DOI: 10.3389/fgene.2019.01110 (cit. on p. 30).

[34] Sanghyun Woo et al. "CBAM: Convolutional Block Attention Module". In: *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 3–19. DOI: 10.1007/978-3-030-01234-2_1 (cit. on p. 30).

[35] Lin Lu et al. "Pancreatic Segmentation via Ringed Residual U-Net". In: *IEEE Access* 7 (2019), pp. 172871–172878. DOI: 10.1109/access.2019.2956550 (cit. on p. 31).

[36] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. DOI: 10.1109/cvpr.2016.90 (cit. on pp. 31, 32).

[37] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. DOI: 10.1109/cvpr.2017.243 (cit. on pp. 31, 32).

[38] Jun Fu et al. “Contextual deconvolution network for semantic segmentation”. In: *Pattern Recognition* 101 (May 2020), p. 107152. DOI: 10.1016/j.patcog.2019.107152 (cit. on p. 32).

[39] Deng-Ping Fan et al. “PraNet: Parallel Reverse Attention Network for Polyp Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 263–273. DOI: 10.1007/978-3-030-59725-2_26 (cit. on p. 32).

[40] Ruxin Wang et al. “Boundary-aware Context Neural Network for Medical Image Segmentation”. In: *ArXiv* (2020) (cit. on pp. 33, 48).

[41] Davide Marzorati et al. “Deep 3D Convolutional Networks to Segment Bones Affected by Severe Osteoarthritis in CT Scans for PSI-Based Knee Surgical Planning”. In: *IEEE Access* 8 (2020), pp. 196394–196407. DOI: 10.1109/access.2020.3034418 (cit. on pp. 33–35, 41).

[42] Xiangrong Zhou et al. “Performance evaluation of 2D and 3D deep learning approaches for automatic segmentation of multiple organs on CT images”. In: *Medical Imaging 2018: Computer-Aided Diagnosis*. Ed. by Kensaku Mori and Nicholas Petrick. SPIE, Feb. 2018. DOI: 10.1117/12.2295178 (cit. on p. 35).

[43] Tsung-Yi Lin et al. “Focal Loss for Dense Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (Feb. 2020), pp. 318–327. DOI: 10.1109/tpami.2018.2858826 (cit. on p. 42).

[44] Hoel Kervadec et al. “Boundary loss for highly unbalanced segmentation”. In: *Medical Image Analysis* 67 (Jan. 2021), p. 101851. DOI: 10.1016/j.media.2020.101851 (cit. on p. 42).

[45] Ken C. L. Wong et al. “3D Segmentation with Exponential Logarithmic Loss for Highly Unbalanced Object Sizes”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 612–619. DOI: 10.1007/978-3-030-00931-1_70 (cit. on p. 43).

[46] Yoni Kasten, Daniel Doktofsky, and Ilya Kovler. “End-To-End Convolutional Neural Network for 3D Reconstruction of Knee Bones from Bi-planar X-Ray Images”. In: *Machine Learning for Medical Image Reconstruction*. Springer

International Publishing, 2020, pp. 123–133. DOI: 10.1007/978-3-030-61598-7_12 (cit. on pp. 44, 46).

[47] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014) (cit. on p. 51).

[48] Malinda Vania, Dawit Mureja, and Deukhee Lee. "Automatic spine segmentation from CT images using Convolutional Neural Network via redundant generation of class labels". In: *Journal of Computational Design and Engineering* 6.2 (2019), pp. 224–232. ISSN: 2288-4300. DOI: https://doi.org/10.1016/j.jcde.2018.05.002. URL: https://www.sciencedirect.com/science/article/pii/S2288430017302464 (cit. on p. 84).