Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

FIIT-182905-110796

Bc. Petra Hlavinová

Development and Validation of Deep Learning Approaches for Detecting Hip Implants from high-resolution X-ray images

Master's thesis

Thesis supervisor: RNDr. Silvester Czanner, PhD.

May 2025

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

FIIT-182905-110796

Bc. Petra Hlavinová

Development and Validation of Deep Learning Approaches for Detecting Hip Implants from high-resolution X-ray images

Master's thesis

Study programme: Intelligent Software Systems

Study field: 9.2.5 Software Engineering

Training workplace: Institute of Informatics and Software Engineering, FIIT STU, Bratislava

Thesis supervisor: RNDr. Silvester Czanner, PhD.

May 2025

Slovak University of Technology in Bratislava
Institute of Informatics, Information Systems and Software Engineering

Faculty of Informatics and Information Technologies
Academic year: 2024/2025
Reg. No.: FIIT-182905-110796

STU FIIT

MASTER THESIS TOPIC

Student:	**Bc. Petra Hlavinová**
Student's ID:	110796
Study programme:	Intelligent Software Systems
Study field:	Computer Science
Thesis supervisor:	doc. RNDr. Silvester Czanner, PhD.
Head of department:	doc. Ing. Ján Lang, PhD.

Topic: **Development and Validation of Deep Learning Approaches for Detecting Hip Implants from high-resolution X-ray images.**

Language of thesis: English

Specification of Assignment:

Approximately 20% of individuals over 40 suffer from bone degenerative diseases, with a rising need for Total Hip Replacement (THR) procedures. Currently, the diagnosis of implant loosening is prone to errors and variability due to subjective human interpretation. With the use of DL models, we aim to provide a more accurate and objective analysis, improving the efficiency and reliability of the diagnostic process. This project proposes developing a Deep Learning (DL) model to accurately detect hip implants from X-ray images, addressing the challenges associated with the manual interpretation of radiological images. The expected outcome is an efficient and reliable automatic detection system that significantly reduces subjective variability and errors, improving the overall diagnostic process for patients undergoing hip implant surgeries. The student will design and implement a DL model to identify hip implant regions and discover the properties of the implant. This project will use saliency map visualisation to ensure the model focuses on relevant regions for classification. Student must validate the model using a separate dataset not involved in the training process and compare the model's performance against manual interpretation in terms of sensitivity and specificity. Finally, student will collect feedback from radiologists and surgeons to further refine and optimize the model.

Deadline for submission of Master thesis:	11. 05. 2025
Approval of assignment of Master thesis:	15. 04. 2025
Assignment of Master thesis approved by:	prof. Ing. Vanda Benešová, PhD. – Study programme supervisor

Annotation

Slovak University of Technology Bratislava
Faculty of Informatics and Information Technologies
Degree Course: Intelligent Software Systems

Author: Bc. Petra Hlavinová
Diploma Thesis: Development and Validation of Deep Learning Approaches for Detecting Hip Implants from high-resolution X-ray images
Supervisor: RNDr. Silvester Czanner, PhD.
May 2025

This thesis deals with the development of an automated system for segmentation of hip implants from X-ray images with an emphasis on the interpretation of epistemic uncertainty.

Such an approach can significantly help radiologists, as it will clearly indicate areas where the model is not completely sure about determining the segmentation mask, and allow them to focus on these areas. As part of the work, we tested several architectures and selected the most effective one for the segmentation itself.

We also included an estimate of epistemic uncertainty in this segmentation, thanks to which we can identify areas where the model is reliable, and conversely, those that should be removed or penalized to make the resulting mask more accurate.

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Inteligentné softvérové systémy

Autor: Bc. Petra Hlavinová
Diplomová práca: Vývoj a validácia prístupov hlbokého učenia pre detekciu bedrových implantátov z röntgenových snímok s vysokým rozlíšením.
Vedúci diplomového projektu: RNDr. Silvester Czanner, PhD.
May 2025

Táto diplomová práca sa zaoberá vývojom automatizovaného systému na segmentáciu bedrových implantátov z röntgenových snímok s dôrazom na interpretáciu epistematickej neistoty.

Takýto prístup môže významne pomôcť radiológom, pretože jasne vyznačí oblasti, kde si model nie je úplne istý určením segmentačnej masky, a umožní im zamerať sa práve na tieto miesta. V rámci práce sme otestovali viaceré architektúry a vybrali tú najefektívnejšiu pre samotnú segmentáciu.

Do tejto segmentácie sme zároveň zapojili aj odhad epistemickej neistoty, vďaka čomu dokážeme identifikovať miesta, kde je model spoľahlivý, a naopak tie, ktoré je vhodné odstrániť alebo penalizovať, aby výsledná maska bola presnejšia.

I honestly declare that I prepared this work independently, based on consultations and using the mentioned sources.

In Bratislava, 11.05.2025 Petra Hlavinová

Special Thanks

I would like to take this opportunity to thank everyone who helped me in the preparation of this thesis. First of all, my thanks go to doc. RNDr. Silvester Czanner, PhD. for his professional project management, valuable advices, always positive consultations and patience throughout the research. His trust in this work and willingness to share his own experiences significantly contributed to the fact that the original idea became a comprehensive and meaningful work.

I would also like to thank doc. Mgr. Gabriela Czanner, PhD., who helped me in the final stages with consultations regarding the quantification of uncertainty and provided useful comments on the interpretation of the results.

I also greatly appreciate the help of MUDr. Libor Nečas, thanks to whom we obtained a high-quality set of X-ray images of hip implants - a key prerequisite for conducting the entire research.

I would also like to thank my friends and classmates for their moral support, mutual knowledge sharing and stimulating discussions, which often helped me find new solutions. Your optimism and shared effort to make studying a pleasant time were a great support to me.

Finally, I express my sincere gratitude to my parents and entire family for their constant support, motivation and encouragement throughout my studies and the preparation of this thesis.

Contents

Abstract

In this thesis, With a focus on measuring the model's epistemic uncertainty using Monte Carlo dropout and entropy metrics, we developed and put into practice a thorough methodology for automatically segmenting hip implants on radiological images (the initial resolution of 3000×2000 pixels was altered to 256×256 pixels). Through systematic experiments, we investigated the impact of varying the number of MC samples (N = 0,10, 20,30,40,50) on segmentation accuracy. We compared the aggregation of results using the mean and median, validated the efficacy of excluding the 10% most uncertain pixels, assessed the influence of training data augmentation, and finally incorporated an entropy penalty into the loss function with adjustable weights. The results indicated that the basic U-Net achieves an average validation Dice score of about 0.766, while the Dice increases to 0.859 ($p<0.001$) when 10% of the most uncertain pixels are removed. Aggregation using the median versus the mean did not show a statistically significant difference ($p > 0.05$), training augmentation did not result in a significant improvement in accuracy but did reduce predictive uncertainty slightly, and the entropy penalty-trained model (optimal weight $\lambda = 0.05$) maintained a robust level of uncertainty while achieving a final validation Dice of 0.902.

Chapter 1

Motivation

According to the World Health Organization (WHO), in 2019, bone degenerative diseases have become much more common worldwide, impacting about 20% of people over 40 [1]. Out of all the available treatment options, Total Hip Replacement (THR) has become a key intervention for many patients, providing relief and regaining their mobility. However, the success of such procedures depends on precise diagnosis, especially when it comes to identifying aseptic loosening, a problem that affects a large number of hip implant users.

Current diagnostic techniques for detecting complications in hip implants heavily rely the subjective and inaccurate manual interpretation of X-ray images. However, this approach is threatened by inherent limitations. Human error and variability can occur in the interpretation of X-ray pictures, leading to diagnostic mistakes that may have major effects on patient care. In this specific situation, the use of Deep Learning (DL) models may prove advantageous.

A lot of discoveries in the medical field in recent years have brought attention to the important role that Deep learning plays. Recent developments in Deep Learning

have demonstrated remarkable success in various medical imaging applications, ranging from detecting malignancies in oncology to identifying abnormalities in cardiology. In addition to speeding up diagnostic procedures, the use of machine learning techniques has brought in a previously unobtainable level of objectivity. It is obvious that an inventive method of implant diagnosis is required. In response to this need, this research seeks to improve the identification of hip implants from x-rays through the use of Deep Learning (DL) models.

In summary, this project stands at the intersection of technology and medicine and represents the potential of Deep Learnig to change the way we approach medical diagnostics. By this work, we seek to improve the health of patients and contribute to medicine in the context of hip replacement by addressing the weaknesses of existing diagnostic approaches. It must be acknowledged, though, that it still aligns with the broader goal of integrating machine learning into medicine, not as a replacement for human expertise, but as a complementary tool that enhances the precision and reliability of medical diagnostics.

Chapter 2

Analysis

2.1 Bone Implants

Thousands of years ago, people learned how to turn clay into pottery. This caused a change that led to great improvements in the quality and length of life. The material that helped preserve grain has become an important part of medicine in the last forty years, improving people's quality of life. More specifically, it is the development of specially designed and manufactured ceramics to repair and reconstruct damaged or "worn out" parts of the body. We call these ceramics bioceramics. Bioceramics are produced in different forms to perform many different functions in the restoration of the body. However, we are interested in its use in the form of specific shaped materials called implants [31].

Bone implants are devices used in medicine to replace damaged or missing bones and joints in the human body. An important part of these implants is that they are usually made of biocompatible materials that are designed to integrate with human tissue as best as possible [52]. The actual acceptance of the implant by its host is not at all obvious. Any material implanted into living tissue elicits a

reaction from the host tissue. It is very important that any implant material avoids a toxic reaction, as this can kill cells in surrounding tissues or release chemicals and cause systemic damage to the patient [31].

Bone implants were first used in antiquity, but modern implants as we know them today only began to develop in the 20th century thanks to important discoveries of new materials and surgical techniques. They have become necessary to improve the quality of life of patients with degenerative joint diseases and bone injuries [55].

2.1.1 Materials of Bone Implants

As a result, bone implants have increasingly spread to several medical specialties. Today they are commonly used in orthopaedics, traumatology and neurosurgery. Every application requires a unique implant selection that takes the patient's demands, options, and risks into consideration. For example, hip replacements must be designed to be able to handle high loads and allow smooth movement.

Based on each application, we can also classify several types of bone implants. Individual implants differ in the types of materials they are made of, their shape, size and function. At the end of the day, choosing the right implant is critical to the success of every surgery and the long-term health of the patients [31].

In terms of material, we distinguish these implants:

1. metal implants (using e.g. titanium, stainless steel)
2. ceramic implants
3. polymer implants
4. composite implants

Although metal implants appear to be the most commonly used in orthopaedic applications, each material has its own advantages and disadvantages. Geetha et al. (2009) in their study reported that titanium is the preferred material for bone implants due to its high strength, light weight and excellent biocompatibility. Because of this, implants are able to provide long-term stability and functionality in the human body, while ceramic implants are valued for their wear resistance[24].

According to Hench and Wilson (2013), ceramic materials such as zirconia and alumina are also a good choice because of their excellent biocompatibility [31].

Polymeric materials such as polyethylene and polymethyl methacrylate (PMMA) are widely used. It is due to their incredible flexibility and biocompatibility. This flexibility means they can be shaped into a variety of forms and are ideal for applications where adaptability is required. An example of their use is the replacement of intervertebral discs. Unfortunately, their disadvantage is that they have a lower resistance [55].

Finally there are Composite implants, which combine the advantages of several materials to improve both their mechanical and biological properties. For example, composites based on carbon fibre and polymers can provide high strength as well as good biocompatibility. According to Hermawan et al. (2010), these are the materials often used in orthopaedics and traumatology for bone and joint replacement [32].

2.1.2 Appearence of Bone Implants

For the purpose of this thesis, it is necessary to be able to distinguish different types of implants according to their appearance. An important fact is that bone implants are clearly visible on the radiograph. On X-ray images, we see them as bright structures that contrast well with the darker surrounding tissue.

Metal implants have a big advantage and that is their high density - radiopacity, which means that they absorb much more X-rays than the tissue next to them. Because of this, metal implants have a significant contrast on X-ray. This characteristic is critical to be able to use implants in cases where clear visualization of the implant during and after surgery is needed [24].Ceramic implants are not lacking and, like metal implants, they also have a high radiopacity [31]. It is meaningful to talk about radiopacity in polymer and composite implants. Their visibility on roentgen images is different. This depends on their composition, some polymeric implants may be less radiopaque, which results in their poor visibility and may appear as grey areas. Fortunately, these can still be modified by adding radioactive components [55].

Furthermore, the identification of implants on radiographs is possible due to the their characteristic shapes and structures, but also according to the size and specific identification marks that the manufacturers add to the implants. As we have already mentioned, every implant is used specifically for a given application. This brings us to the fact that individual implants have different shapes, based on what part of the human body they are supposed to replace. For example, as we can see in the figure 2.1 below, hip Implants have a specific shape that includes a ball head and neck, while knee Implants have a shape that matches the anatomy of the knee [32].

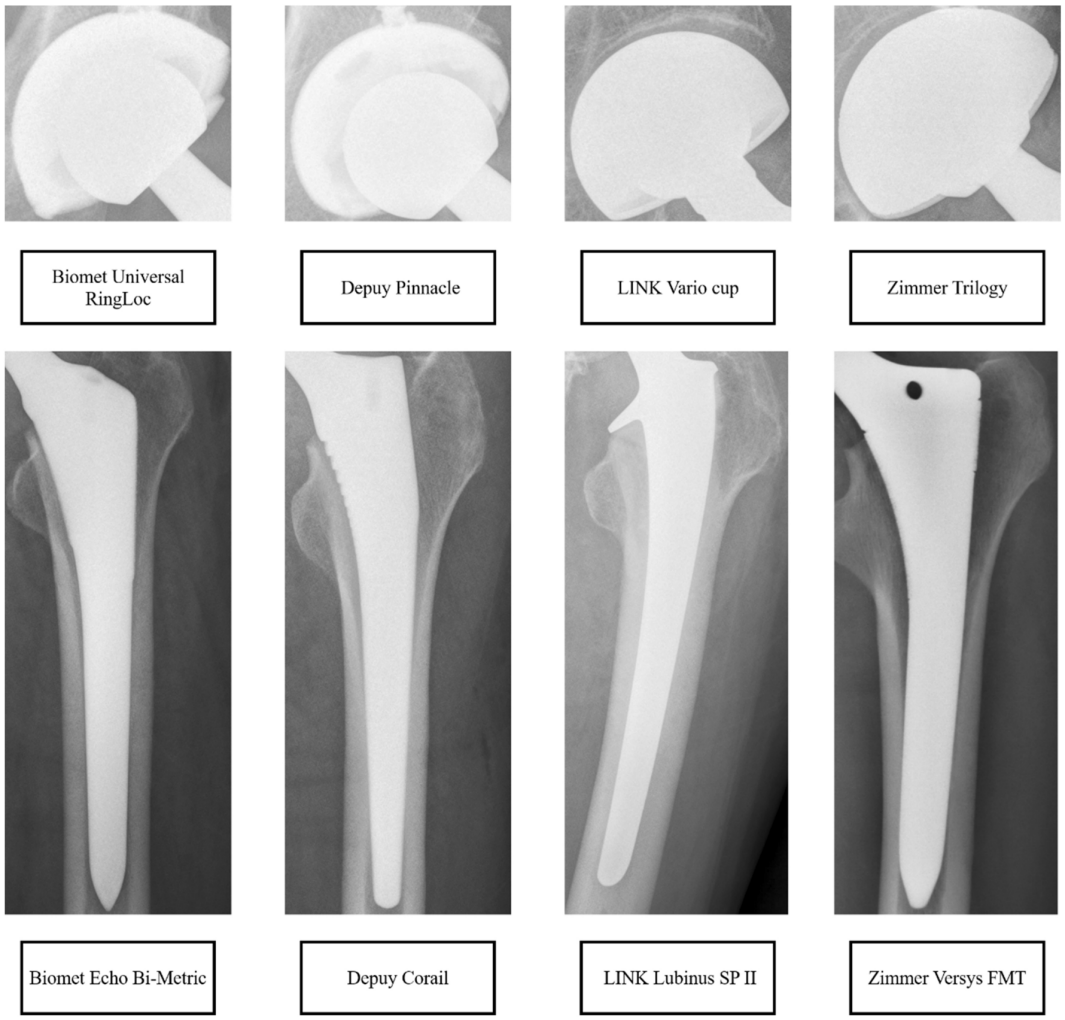


Figure 2.1: Different types of hip Implants on X-rays. Each implant labelled with the manufacturer's name and specific model. We can see the cups in the the upper part of the picture and the stems in the lower part [25].

Depending on the need, they replace or repair damaged tissues, bones or joints in different parts of our bodies.

1. **Dental implants** Dental implants are used when patients want to have their missing teeth replaced. This implants are screws, most commondly made of titanium, that are inserted into the jaw. Then they serve as an anchor for

new teeth or bridges [55].

2. **Knee implants** Knee implants are used for knee joint replacement, not only but mostly during the cases such as osteoarthritis. The knee joint is a complex structure that must cope with a critical load during life when performing various daily physical activities, such as walking, sitting, bending, running. Knee replacements can be done by two methods: Total knee replacement and Partial knee replacement. The goal is to remove all defective and damaged joint parts and surfaces and replace it with an implant [41][8]. The knee joint itself consists of 4 parts: femur, tibia, patella and meniscus and total knee replacements are made of these four parts as well:

   (a) Femur - the top part

   (b) Tibia - shin, the bottom piece

   (c) Liner - fills the space between metal pieces as a plastic cartilage

   (d) Patella - also named knee cap, is only optional part and it is not always used in the replacement [33]

3. **Spinal implants** Spinal implants include intervertebral plates, screws, and rods that secure and support the spine. Intervertebral plates replace damaged or degenerated discs, while screws and rods are there for stability to support the spine and allow it to align properly. [32][66].

4. **Shoulder implants** Shoulder implants are used to replace damaged shoulder joints, again most often with arthritic changes or after injuries. Shoulder joint is very complex joint too, so the implants have to be designed to provide maximum functionality and comfort for the patient. Implant like this consists of two parts: the humeral component and the glenoid component [11].

5. **Elbow implants** Elbow implants are used to replace damaged elbow joints. An important element of this joint is its bending and rotation. Implants must be able to imitate these movements. Therefore, this replacement is performed through two parts, the humeral component and the ulnar component [55].

6. **Hip implants** It should be noted that although bones have the ability to self-heal the surrounding tissues, some bone damage is permanent and irreversible. This happens often in bone diseases such as osteoporosis. However, the problem can also arise with various injuries. Unfortunately, hip joint injuries are a very common occurrence. What is worse is that it is a serious condition that can even lead to permanent disability and death. Even if it is not such an escalated case, patients with hip joint disease have problems performing everyday activities. This is the reason behind the annual increase in replacement of hip bones [28]. Hip implants are typically made of four individual components:

   (a) Femoral Stem

   (b) Acetabular Cup

   (c) Femoral Head

   (d) Acetabular liner

   For a better overview of what hip implants look like, see the picture 2.2 at the end of this section.

   Femoral Stem is the part that is insertet into the femur od thing bone. This is the part, where the artificial joint is placed on top. Due to the long-term results, stems are typically made of titanium and cobalt-chromium. Because these materials are well tolerated by human living tissue.

Acetabular Cup fits into pelvis. Its task is to hold the plastic liner of the hip replacement. It also serves as a fake cartilage, but in some cases the plastic liner is used without this metallic cup.

Femoral Head also called ball fits on the end of the stem. According to American Association of Hip and Knee Surgeons. End of the stem where we put the head is shaped taper which enables the head to be wedged into place and be firmly kept. The diameter of the head varies, which is frequently affected by the size of the cup that slides into the pelvis.

Acetabular liner is the most weak part of total hip replacement. It is there to replace the real cartilage, but is susceptible to be wear-out. This part is placed into the cup [30] [33].

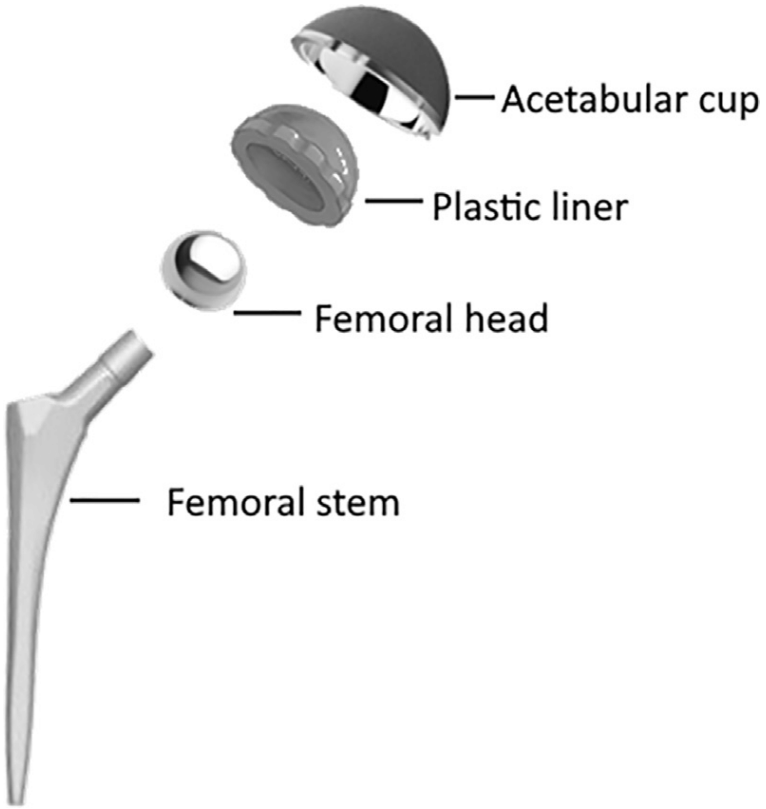


Figure 2.2: Parts of hip implant [53].

2.2 X-ray images

In 1895, Wilhelm Conrad Röntgen detected electromagnetic radiation in the wavelength range, which is known today as X-rays. The name X-rays should emphasize their unknown nature. X-rays is a specific kind of high-energy electromagnetic radiation that are used to make images by passing X-rays radiation, produced by x-ray tubes, through the tissues of the human body and catching the radiation on a film or digital detector to create a picture [13]. After that, X-ray images produce two-dimensional representation of the body's structure, with denser tissues, such muscles and organs, appearing darker and less dense materials, like bone, absorbing more radiation and appearing lighter on the picture. This discovery quickly initiated intense research in a number of areas and led to the creation of radiology[29]. A year later, in June 1896 was treated by radiotherapy the first patient [64].

X-rays images are essential in the diagnosis and treatment of many medical conditions. They are used on daily basis for diagnosis of fractures and other bone injuries, identifying lung diseases such as pneumonia or lung cancer, evaluating the condition of teeth and jaws in dentistry or detection and monitoring of joint and bone abnormalities, including the presence of implants.

Due to these beneficial attributes for medicine and the gradual development of digital technologies and computing, there is an increasing interest in automating the analysis of X-ray images using neural networks. For there purposes, convolutional neural networks that are widely used in image recognition and processing. CNNs are capable of learning from visual data and extracting intricate details and therefore, they can be trained to recognize anomalies in X-ray pictures, such as fractures, tumors, or foreign objects. As an example of the use of x-ray images in neural networks, Rajpurkar and his colleagues developed a model that can detect

pneumonia from chest X-rays, which achieved high accuracy even comparable to practicing radiologist. To achieve that by using 121-layer CNN that was trained on the largest available chest x-ray dataset, at the time.

2.3 Image Preprocessing

This study's data collection procedure consist of obtaining a wide variety of hip implant X-ray images. Nonetheless, issues like inconsistent image quality, disparate imaging protocols, and patient privacy concerns need to be taken into account and resolved. It is crucial to ensure and standardize image format of every image for consistent model training and evaluation [19].

When employing neural networks for medical imaging, image preprocessing and segmentation are essential steps in the process, especially when identifying various implant types from X-ray images. The preparation and processing of the input images has a significant impact on how well a neural network recognizes and analyzes these implants. Because if we put data into the neural network that are not good enough, we can never expect a good enough result, regardless of how perfect a model we create.

In the context of X-ray images, image processing refers to a variety of methods for improving and modifying images in order to enable effective analysis. This consist of image normalization, augmentation, contrast improvement and noise reduction and edge detection citelecun2015deep.

2.3.1 Image Normalization

Image normalization is a technique of bringing an image's pixel values inside a common range. Usually, it consist of scaling the pixel values to a standard de-

viation of one and a mean of zero. which aids in the consistent performance of the neural network across different images. It reduces the impact of variations in image intensity and illumination [35].

This process ensures that the models can learn and generalize effectively by eliminating variations caused by different lighting, contrast, and other factors. By using this technique we ensure that all images in the dataset have a consistent range of values, which facilitates model training and improves model stability [50].

2.3.2 Noise Reduction

Another problem with our data is noise. Noise reduction is a key technique in image processing. The aim of the technique is to improve the quality of images by removing disturbing elements caused by various types of noise. Noise can be caused by various factors, such as low lighting intensity, or other disturbing elements in the environment. Noise reduction is important to improve the quality of images and is an important part that is performed before processes such as edge detection, segmentation and classification [26].

Noise in X-ray pictures can make it more difficult for the neural network to interpret the data correctly. Low radiation dosages, patient movements, and equipment constraints are a few possible causes of these distortions. To reduce noise or undesired artifacts in X-ray images, noise reduction techniques are applied. Gaussian filtering, and median filtering are common techniques for reducing noise. By boosting the image's clarity and lowering its susceptibility to misunderstanding, noise reduction increases the precision of diagnosis. [10].

2.3.3 Augmentation

Augmentation is the process by which we obtain new artificial data. Since the use of machine learning involves working with lots of data, it can happen, for example, that our neural network cannot train sufficiently and thus learn to recognize patterns. An amount of data is essential in many applications of machine learning. Some tasks require several thousand inputs. Fortunately, if we don't have dataset that is large enough, we can make him a bigger one. Thanks to augmentation, we can increase the accuracy and robustness of the models. In addition, it increase the diversity of training data, which reduces the risk of overfitting and increases the ability of models to generalize to unseen data.

There are several ways we can augment data - create new data. Basic techniques include rotating and flipping. We create new variations by rotating the image by different angles or by mirroring it horizontally or vertically. Furthermore, we can scale the data to different sizes or crop it. Moving pictures horizontally or vertically using translation helps us our model to focus on more important parts of the pictures. In addition, we can add noise, which improves the robustness of the model against noise. Random changes in brightness, contrast, saturation, and hue can also help models handle data in different color conditions [61].

When we take the medical background into account, the process of data augmentation entails modifying the original X-ray pictures in order to produce more training examples. As a result, the training dataset is more diverse and the deep learning model is more resilient to changes in patient positioning, imaging conditions, and other variables.

2.3.4 Smoothing

Smoothing is a technique, which is used to blur an image. By this we can remove line artefacts, segment edges and overall improve the quality of the image. This is crucial process before further analysis such as edge detection.

To create a smooth image blur we commonly use a **Gaussian function**, which remove line arteffacts and noise while preserving details of the picture [26].
Another method of smooting is **Bilateral filtering**. It reduces noise and retains edges by combining radiometric and spatial data [63].
Another technique is **Interpolation**, this is used to increase image resolution and improve quality after smoothing. Its methods calculate new pixel values between existing pixels, increasing the smoothness of the image due to smooth transitions [36].
The last method we will mention is **Anti-aliasing pixels**. This metric is important in our work, as it is very useful in the visualization of medical images. We use it to soften the pixel edges and reduce artifacts (aliasing). This aliasing appears as jagged edges on the edges of objects, which is caused by insufficient image resolution. There are several ways to do anti-aliasing, one of them is to generalize the whole image in high resolution and then reduce the size to the size we require. By anti-aliasing pixels we make the pictures more natural and less distracting. [45].

2.3.5 Edge Detection

When processing images in computer vision, we come to a situation where we need to analyze the image. The edges of individual objects help us to find out what is on it. Edge detection is a commonly used process that tries to record significant parts of objects in the image. Image derivatives are used for these purposes, however,

these derivatives are very sensitive to noise sources. In order to eliminate this problem, the images should be smoothed. However, you should pay attention to the fact that smoothing can cause some loss of information.

Another important fact is that there is no universal algorithm for edge detection that would work for all purposes.

Let's first define edge. Consider them as points in the image where the intensity changes significantly. Or, on the other hand, we can see them as discontinuities of scene objects. The obtained edges are important for segmentation and identification of structures. [70]

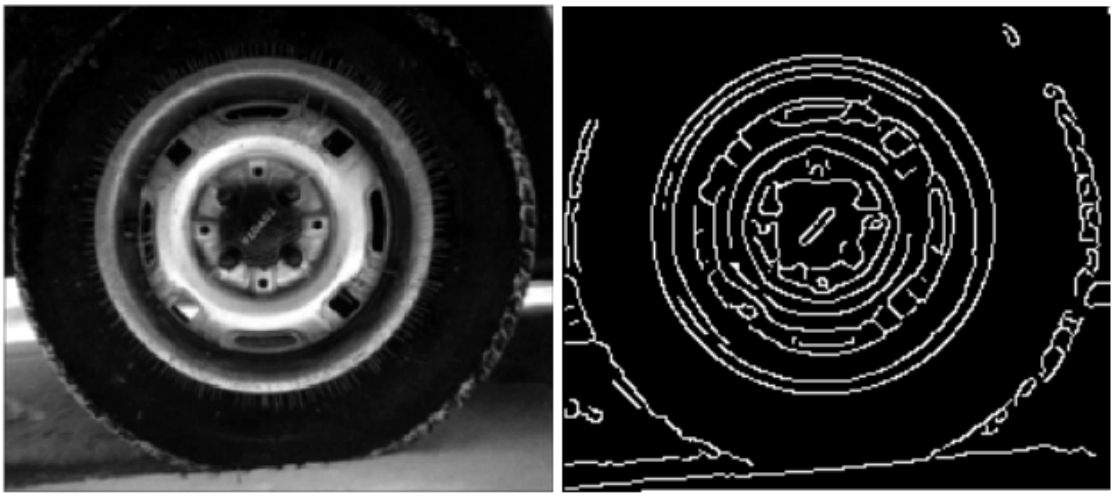
There are two edge detection techniques:

- Linear methods

  These methods use image convolution with linear filters. Among the most famous linear methods are Sobelov filter or Prewittov filter. Advantage of these methods is that there are simple and fast. On the other hand, they can be sensitive to noise.

- Non-Linear methods

  Non-linear methods, on the other hand, use advanced algorithms that take into account the local characteristics of the image, which enables better handling of noise and more accurate edge detection. An example of this method is Canny edge detector. This method involves a multi-step process, which results in admirable accuracy and robustness in edge detection, including image smoothing and non-maximum suppression [14]. Usage of this method we can see in the picture 2.3 below.



(a) (b)

Figure 2.3: Edge detection experiment for tire image, (a) tire image, (b) traditional Canny's edge detection algorithm [56].

2.4 Machine Learning

Machine learning (ML) is a field of artificial intelligence (AI) that focuses on the development, study and application of statistical algorithms enabling computers to learn from and make predictions or decisions based on data. This is an major change from traditional rule-based algorithms as it may accomplish a goal without being specifically designed to do so. Machine learning has attracted a lot of interest recently from a variety of fields, such as technology, healthcare, and finance, because of its capacity to handle massive amounts of data and reveal patterns and insights that are frequently invisible to human study [46].

The fundamental idea of machine learning is to build models that can make generalizations from the data they used for training to retrieve new, unknown input. This is achieved through a process called training, when an algorithm is exposed

to a large dataset and learns from it. The quality and quantity of data play a critical role in the effectiveness of Machine learning models. For example, the fast progress and implementation of machine learning (ML) in the healthcare industry has been made possible by the availability of large datasets, including genetic data, medical pictures, and electronic health records [48].

Machine learning can be broadly categorized into three types:

1. Supervised Learning

   Supervised learning involves training a model on a labeled dataset, where the desired output is known. This method is frequently applied to classification and regression problems, such forecasting patient outcomes or diagnosing diseases from medical images

2. Unsupervised Learning

   Unsupervised learning deals with unlabeled data, which allows it to discover underlying patterns, groupings, or associations in the data. This may be essential for figuring out new medical problems or patient demographics.

3. Reinforcement Learning

   Reinforcement learning is a type of Machine Learning where an agent gains decision-making skills by acting in a way that increases a hypothetical cumulative reward. This technique may be used in personalized medicine to optimize strategies for treatment depending on the reactions of specific patients.

This method has potential applications in personalized medicine, where treatment plans can be optimized based on individual patient responses [62].

Machine learning techniques are being used more often in medical imaging to

increase the precision of diagnoses, automate repetitive activities, and offer insights that are not possible with human knowledge. Medical imaging has undergone a revolution thanks to the use of Machine Learning, which has significantly improved illness diagnosis and therapy planning [47].

Machine learning application in medicine has been life-changing. Its algorithms have been successfully applied in drug discovery, genomics, diagnostic imaging and patient care management. In recent years, machine learning was used to precisely indentify and categorize tumors in radiology images, forecast a patient's risk of developing various diseases, based on genetic data. Moreover, treatment plans may be improved by taking into account clinical and patient histories. The results of these developments not only lead to increased accuracy and efficacy in medical diagnosis and treatment, but they also open the way for personalized medicine, which considers the individual needs of each patient when developing treatment plans and regimens [20].

To sum up, machine learning is an exciting step in the development of artificial intelligence and its use in many industries, including healthcare. Improving diagnosis and treatment planning are greatly impacted by its capacity to absorb and learn from massive volumes of data, spot trends, and make predictions. But in order to successfully implement ML in healthcare, issues with data quality, model interpretability, and workflow integration must be resolved.

2.4.1 Neural Networks

One of the key elements of DL are neural networks, or NNs. These networks consist of networked nodes, or neurons, that process information and then send signals to other neurons. Every neuronal connection has the ability to transfer signals to other neurons. After processing the signal, the receiving neuron notifies neurons that are below it.

NNs consists of three main types of layers:

1. an input layer
2. one or more hidden layers
3. an output layer

The input layer is the place where the network's initial features or data are received. The input data is processed by the Hidden Layers, by using a series of weighted connections and non-linear activation functions. The term "hidden" refers to the fact that they cannot be observed directly in the input or output. Finally, the network's predictions or outputs are generated by the last output layer. Every neuronal connection has a weight attached to it. The strength of the connection is determined by these weights, which are gained during the training process[27]. Training a neural network involves this iterative feedforward and backpropagation process. After the feedforward process - passing input data through the network's layers to produce predictions or outputs, the network's output is compared to the actual target values. The error between the predicted output and the actual target is calculated using a loss function. The process of propagating this error backward through the network to update the weights, is called backpropagation and it makes neural networks learn and become more efficient. This iterative procedure keeps on till the performance of the network becomes high enough [59],[43].

The essential part of training a neural network is tuning of hyperparameters - settings or configurations that we specify before training begins. It involves learning rate, batch size, and the choice of activation functions. Hyperparameter tuning involves systematically exploring different combinations of hyperparameters to find the optimal set that results in the best model performance on a validation dataset [43].

There are several neural network topologies, each intended for a particular set of applications and data kinds. In medical imaging, Convolutional Neural Networks (CNNs) have been successfully applied to analyze various types of images, including X-rays, MRI scans, and CT scans. For instance, CNNs may be trained to identify anomalies with a high degree of accuracy in the detection of diseases like cancer. They work especially well in scenarios where the patterns that need to be recognized are modest and could be challenging for human radiologists to recognize. Large-scale dataset collection and autonomous learning make CNNs an effective tool for improving medical diagnosis [44].

2.4.1.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a specialized type of neural network used for processing data that has a grid-like topology, such as images. CNNs are exceptionally good at problems including object identification, segmentation, and image classification because its architecture is made specifically to identify and extract patterns from visual input.

Convolutional layers, which apply filters to the incoming data, are the main component of CNNs. These filters are tiny receptive fields that scan the input image and multiply the filter values by the original image pixel values in order to execute convolution operations. Feature maps that emphasize particular aspects of the pic-

ture, such edges, textures, or particular objects, are produced as a consequence of this procedure. CNNs may learn progressively more complicated features at each layer by stacking numerous convolutional layers with pooling layers in between to lower the spatial size of the representation. These characteristics are used by the final fully linked layers to categorize the picture. [40].

In conclusion, neural networks, especially CNNs, have completely changed the fields of artificial intelligence and machine learning. Many disciplines have benefited much from their capacity to handle and learn from complicated data. One of the most well-known sectors that benefit substantially from this technology is medical imaging. It is believed that as neural networks grow, they will play an increasingly significant role in healthcare, leading to improvements in patient care, diagnosis, and treatment planning.

2.4.2 Connection between X-rays and CNN

As we stated before X-ray images are very important tool for medicine diagnostics, because they are providing a detailed view of the internal structures of the body. The fact that X-ray images really are images, it offers us to process it using convolutional neural networks.

Their involvement in this field made a revolution in the analysis of medical images. This speeds up and increases the accuracy of the diagnostic process, because the trained model is deployed in clinical practice, where it automatically analyzes new X-rays and can provide diagnostic predictions [44]. What is interesting, is that CNNs are achieving high accuracy in detecting pathological findings and even often outperform human experts [54].

Classification, localization and segmentation are the most common tasks for which neural networks are used in the processing of X-ray images. For example, we can

determine the position and shape of structures or abnormalities. As an example of the use of CNN, Ronneberger et al. (2015) proposed a U-Net architecture that is widely used for the segmentation of medical images, not only but also including X-ray images [57].

However, these are far from the only cases of successful applications of neural networks to X-ray images. Another great application can be attributed to Zimmermann and his colleagues (2019), who used CNN to detect bone fractures. Their model detects fractures of the distal part of the forearm even with high accuracy [67].

One of the most prominent uses of X-ray CNNs is the diagnosis of lung diseases such as pneumonia, tuberculosis and COVID-19. Thanks to these uses, we obtained models that were able to detect pneumonia from X-rays of the chest at the level of radiologists (CheXNet model) [54], models that achieved great accuracy in the detection of tuberculosis [42], or models used to detect COVID-19 that brought fast diagnosis during a global pandemic [68]. An example of Rajpurkar's model's results can be seen in the picture 2.4 below.

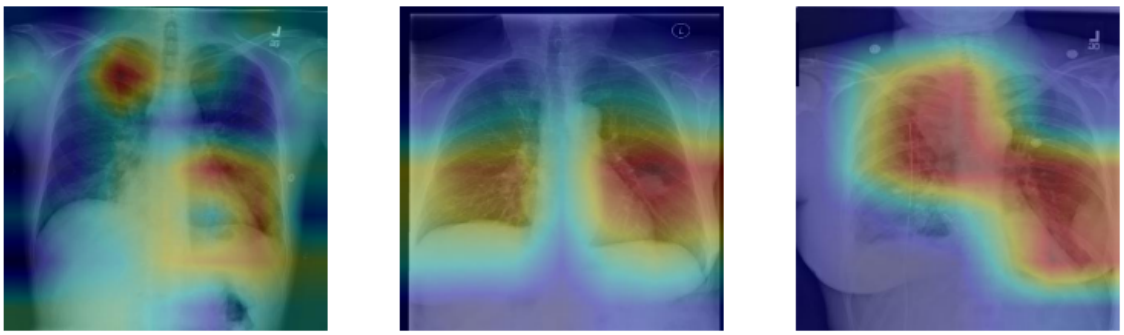


Figure 2.4: In the provided images, we can see the results of the CheXNet model, developed by Rajpurkar et al., applied to chest X-rays for the detection of pneumonia, where in all three images was the pathology correctly classified [54].

Many relevant studies supports the use of these methods to identify implants and

evaluate their placement. We can mention some approaches that offered reliable and accurate methods for detecting dental implants in X-ray images, providing potential benefits for early diagnosis and treatment planning: A research that achieved a classification accuracy of 71.7% suggested a unique approach based on two-dimensional projections of three-dimensional surface models. This research described the use of synthetic X-ray images for deep learning-based dental implant identification [18].

Another study demonstrated that hip implant loosening could be detected with 96.11% accuracy by using deep convolutional neural networks (DCNNs) and a stacking classifier. The study constructed a deep learning approach to identify the implant location in hip X-ray images using the YOLOv5 architecture [34].

In conclusion, the use of ML in processing x-ray images for detecting hip implants is a highly promising area of research. By combining modern computing technologies with medical imaging, this strategy has the potential to greatly improve the health of patients and diagnostic accuracy.

2.5 Segmentation

Segmentation plays a vital role in medical diagnosis using X-ray images. It helps to detect and analyze fractures, lung diseases, and other ailments.

Segmentation in deep learning means the process of partitioning a digital image into multiple segments to simplify the image and make it easier to understand and evaluate. It is the process of dividing the image into segments or regions of interest (ROIs). In the context of X-ray images, segmentation is crucial for isolating specific areas like bones, organs, or abnormalities. For detecting hip implants, segmentation focuses on isolating the implant from the surrounding bone and tissue.

The reliability of the diagnosis and the ensuing treatment planning are directly impacted by the segmentation accuracy. To precisely define the implant, sophisticated methods like region-based segmentation, edge detection, and thresholding are used [7],[51].

Segmentation can be performed manually by human annotators or through automated methods, such as Convolutional Neural Networks (CNNs) and U-Net. Besides that there is also a combination of these methods called semi-automatic segmentation.

Manual segmentation In manual segmentation, skilled human annotators carefully delineate the boundaries of objects or regions in an image using specialized software tools. This process can yield exact findings when performed by specialists and is extremely accurate. Nevertheless, it is time-consuming, labor-intensive, and often not suitable for large-scale datasets.

Automatic segmentation On the other hand, the other option for automatically segmenting images is using CNNs. Based on sizable labeled datasets, these deep learning algorithms are able to identify and distinguish objects or regions in images. Even a specific CNN architecture for biomedical image segmentation was developed, U-net. U-net is made up of a symmetric expanding path for accurate localization and a contracting path for feature extraction. Although U-Net can be modified for a variety of applications, it is especially well-suited for biomedical image segmentation tasks [57].

Semi-automatic segmentation Semi-automatic segmentation combines automatic techniques with user input. This approach is particularly useful in cases

where complete automation is difficult or where high accuracy is required, such as in medical image analysis.

We can perform this method with different approaches:

- **Interactive thresholding** Using interactive thresholding The user needs to set threshold values and visually evaluate segmentation results in real time, thereby increasing the accuracy of the process [49].
- **Interactive growth of regions** In this case, the user does not select a threshold, but selects seed points in the image, and the algorithm itself automatically expands regions based on these points and similarity criteria [3].
- **Active contours (Snakes)** Active contours are curves that move under the influence of internal forces (ensuring the smoothness of the curve) and external forces (attracting the curve to the edges of objects) [37].
- **Graph Cut** Graph Cut is a method that uses a graph representation of an image, where nodes represent pixels and edges represent similarities between them. The task of the user in this case is to mark some pixels as 'foreground' and 'background', and the algorithm will then find the optimal division by itself [12].

2.6 Uncertainty in AI

The practical implementation of artificial intelligence (AI) systems in clinical settings requires not only high accuracy but also transparent information regarding the reliability of each output. Neural networks, as deterministic approximators, cannot quantify the certainty of their predictions without additional mechanisms. Together with explainability and accuracy, this is conceptually the 'third pillar' of evaluation alongside accuracy and explainability, providing key information for

subsequent physician decision making, active learning, or adaptive data acquisition [9] [23].

Several systematic review studies have been published that categorize approaches to uncertainty estimation into two general categories: epistemic (model) uncertainty and aleatory (data) uncertainty. In contrast to aleatory uncertainty, which is inevitable due to the fundamental stochastic character of the measurement (noise, patient movement, protocol changes), epistemic uncertainty is a reflection of the limitations of the training data and can be reduced by augmentation [38]. In practical applications, it is therefore recommended to simultaneously estimate both components, particularly in medical segmentation, where an erroneous lesion boundary can result in incorrect treatment [2] [65].

Regarding the aleatory component, two distinct strategies have been proposed in recent literature:

1. direct learning of the parameters of the prediction distribution within the network, known as learned loss attenuation

2. test-time augmentation (TTA). In TTA, a set of stochastically transformed inputs is generated during inference, and the uncertainty is inferred from the variance of their outputs[22].

The most widely used approximation to epistemic uncertainty is Monte Carlo-dropout, which elegantly obtains the distribution of outputs without interfering with the training process of by using dropout layers as a sampling from the a priori distribution of network weights. To put it another way, we insert the dropout into the testing part and perform some sampling.

In the following subsections, we discuss the two main types of uncertainty - epistemic and aleatory - and provide detailed descriptions.

2.6.1 Aleatoric Uncertainty

'Aleatoric' comes from the Latin alea, which means 'a die'. Thus, the aleatoric uncertainty is the type of unpredictability that is inherent in the environment; it is noise that cannot be eliminated, regardless of how much data is gathered or how well your model works.

As was mentioned, The alatoric uncertainty expresses the randomness inherent in the data itself. In medical imaging, the primary source of noise is typically detector noise, varying patient positioning, different exposure settings, or anatomical variations among individuals. Even if we had an ideal network and an unlimited supply of training images, these random variations would persist in the prediction. Consequently, it is widely acknowledged that aleatoric uncertainty cannot be completely eliminated by accumulating additional data; rather, it can only be estimated and appropriately written into the result [38].

The first approach is Test-Time Augmentation (TTA). In 2019, Wang et al. proposed an elegant yet architecturally simple strategy based on (TTA) [65]. The concept is intuitive: If an image undergoes random geometric and photometric modifications during execution, we can artificially replicate these transformations during inference, thus observing the behavior of the network under various variations of the 'same' image. During inference, we generate a set of subtle yet realistic transformations to the original X-ray image. These transformations may include rotations of a few degrees, horizontal flips, or the addition of low-level Gaussian noise. Each variation is then submitted to the same deterministically fitted model, transformed back to the original coordinates, and then finally computed as the average probability map. If the transformations are insignificant for the network, the individual predictions remain identical, and the entropy of the average - the uncertainty remains low. Conversely, if small changes in the input result in distinct

masks, the entropy increases, indicating high aleatory uncertainty. Therefore, the method directly quantifies the stochastic factors that are intractable from the data, without the need for any additional weights [4].

The second technique used is learned loss attenuation, often referred to as the heteroskedastic model. The network simultaneously predicts the log-variance of $\log \sigma^2$ for each pixel in addition to the normal logit μ. During training, the loss is modified so that probabilistic sampling $\mu + \sigma z$ with noise $z \sim \mathcal{N}(0, 1)$ minimized the segmentation error rate. In practice, this causes the model to "inflate" the estimated σ^2 and admits a higher aleatory uncertainty, while in homogeneous zones it pulls it down to almost zero. The value of σ^2 is used directly as a quantitative measure of confidence after training; low variance implies a stable, high variance a potentially inaccurate prediction. The advantage is that this is learned along with the main task and no external sampling is needed, saving time in bulk clinical inference.

2.6.2 Epistemic Uncertainty

The word "epistemic" comes from the Greek word 'epistēmē' which means knowledge. Thus epistemic uncertainty is the "I don't know yet" aspect of the unknowns.

Whereas aleatory uncertainty represents irreducible noise in the data, Epistemic (or "model") uncertainty represents uncertainty in the neural network's own parameters. It appears anywhere the training set is small or does not represent all clinical scenarios. The epistemic component would theoretically decrease to zero if there were an infinite number of radiographs with perfect annotations. However, in reality, it remains significant and determines whether the model can accurately generalize to new implant types or non-traditional projections [23].

According to Alamleh et al., Monte-Carlo dropout emerged as the most robust estimator of epistemic uncertainty. The idea is simple: we keep the dropout layers active during inference, randomly 'switching off' a different subset of neurons at each pass, obtaining the full set of segmentation maps. In the case of using variance as a metric, from the probability set $P_1, \ldots, P_N$, we then compute $\mathrm{Var}_{\mathrm{epi}}(x) = \frac{1}{N}\sum_{n=1}^{N}\left(P_n(x) - \overline{P}(x)\right)^2$, $\overline{P}(x) = \frac{1}{N}\sum_n P_n(x)$,, where the variance represents the extent of the model's uncertainty in a specific pixel [4]. We can see how it is used in architecture when used with Unet in the figure below 2.5.

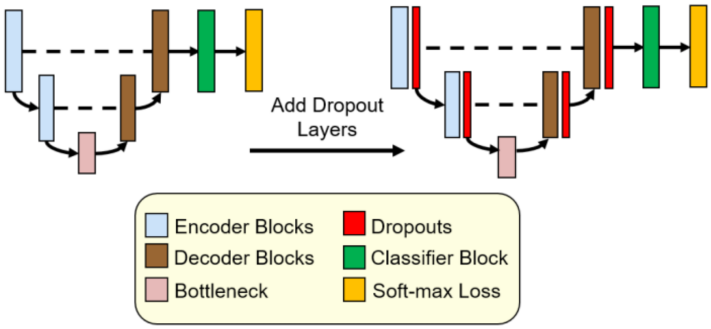


Figure 2.5: The architecture of U-net after adding dropout layers [58].

Neural ensembles are another method, a collection of independent characteristics is produced by training many networks with various initializations and maybe separate training batches $f_1, \ldots, f_N$. The average of their outputs serves as the prediction, while the variance between the branches serves as an estimate of the epistemic uncertainty. The authors demonstrated that ensemble and MC-dropout yield comparable spatial distributions of high variance values. However, ensemble is significantly more computationally intensive, necessitating the separate storage of weights for each term and linearly increasing the inference time.

The actual literature agrees that both Monte-Carlo dropout and neural ensembles are among the most reliable and user-friendly techniques for quantifying epistemic

uncertainty in deep learning. Meta-analyses demonstrate that MC-dropout offers comparable uncertainty calibration to a small (3-5-member) ensemble for medical segmentation tasks with limited data, but at a fraction of the computational expense [4]. In our work, we therefore chose to use MC-dropout as the primary estimator of epistemic uncertainty.

Chapter 3

Solution Proposal

In this chapter, we will focus on the solution proposal for the detection of hip implants using deep neural networks (CNN). Our goal will be to create a convolutional neural network that will automatically recognize the presence and segment the implant on provided X-ray images.

The **input** for our application will be X-ray images that were provided to us by the hospital in Martin, Slovakia. As an **output** of this application, we want to get segmentation of the implant used on X-ray image. For evaluation purposes, the values of the prediction metrics results will also be part of the output.

To achieve this goal, a complex pipeline will need to be used. The proposed pipeline provides an approach to the detection of hip implants using convolutional neural networks. Using various preprocessing techniques, augmentation and training configurations, we aim to optimize the accuracy and robustness of the models. The following diagram 3.1 shows the entire pipeline process from image reception to model output.

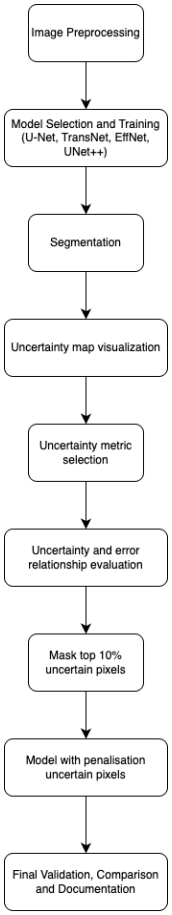


Figure 3.1: Our solution proposal pipeline.

We will need our data to go through important image-processing, which will get our data into the required shape. We will then pass this data to our convolutional neural network. Our neural network will then be trained on this data. In order to compare the results and verify if our neural network is doing what it should the evaluation is needed. For these purposes, we will use evaluation metrics such as accuracy, precision, recall, f1-score. To ensure the best results of our convolutional neural network, we will try several configurations.

The pre-processing of the images, which will be necessary for the purposes of our project, will include automatic cropping focused on the hip joint area using

annotations. Then, it is necessary to reduce the images to a uniform size (256x256 pixels) in order to obtain consistent inputs to the model. Because normalizing input for neural networks is a crucial preprocessing step, it will be needed, where we will use Min-Max normalization or Z-score normalization to improve model convergence. Depending on the quantity and quality of the data we obtain from the hospital, we will also consider using augmentation using techniques such as rotation, mirroring, scaling and adding noise to increase the variability of the data.

Lastly, when the model is trained, we will assess its performance on a held-out test set using Dice score. In order to determine the optimal trade-off between segmentation quality and computational efficiency, we will analyze several network designs and hyperparameter setups. To promote repeatability and direct future advancements, all tests and findings will be recorded.

Chapter 4

Aim and Objectives

This project's main goal is to construct a powerful Deep Learning (DL) model, namely a Convolutional Neural Network (CNN), for the precise segmentation of hip implants using X-ray images. This project aims to solve the difficulties currently associated with identifying illness associated with hip implants, which frequently depend on subjective evaluation and can produce inconsistent outcomes.

4.1 Project Task Schedule

Analysis and Planning

In order to get there, we need to first make a research and analysis. Where we will identify the state of medical imaging as it relates to the diagnosis of diseases associated with hip implants by conducting in-depth research. This involves looking over previous research, identifying the weaknesses of the diagnostic techniques used now, and determining where DL models may be able to help.

Data collection and Preprocessing data

After that, we need to create a large, varied dataset of X-ray pictures of hip

implants. The focus will be on building a dataset that includes a wide range of cases, including early and advanced stages of aseptic loosening. To prepare these images for input into the CNN, preprocessing involving noise reduction, contrast enhancement, and normalization will be essential.

Model Development

The core of this project is designing and implementing a CNN model that can accurately segment hip implant from X-ray images. This involves selecting appropriate architectures, tuning hyperparameters, and training the model with the preprocessed dataset.

Testing and Validation

Testing and Validation involves assessing how well the model performs in comparison to conventional diagnostic techniques. As part of this, we will have to choose and select the right hyperparameters that maximize our dice score. A different dataset that was not used for training will be used to validate the model. In addition, we will test various combinations regarding epistemic uncertainty.

Documentation and Reporting

At the end, we have to document the entire process, from the initial research phase to the final model development and testing. This will include detailed reporting on the methodologies used, challenges encountered, results obtained, and the implications of these findings for the field of medical diagnostics.

The following strategy will guide the steady implementation of the abovementioned actions:

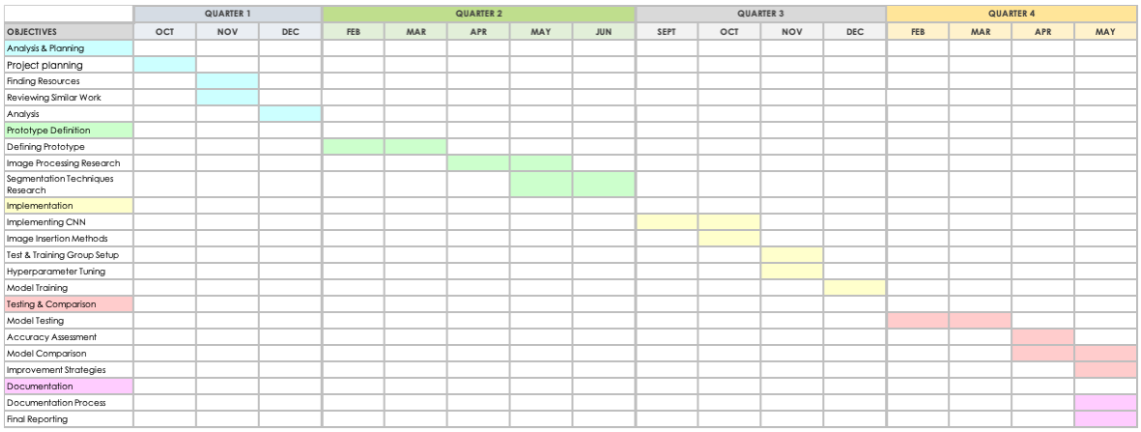


Figure 4.1: Gantt chart

4.2 Research Questions

To configure a convolutional cneural network to recognize different types of implants from X-ray images, the following considerations are crucial:

- **RQ1: How can a U-Net convolutional network be designed and trained to reliably segment a hip implant in standard radiographic projections?**
- **RQ2: Which epistemic uncertainty metric (variance, entropy, IQR, 95%-range) best correlates with pixel-wise segmentation error in Monte-Carlo dropout?**
- **RQ3: How does the addition of an entropy-based penalization term affect the segmentation accuracy (Dice) and the level of predictive uncertainty relative to the baseline model without penalization?**
- **RQ4: Can removing the most uncertain pixels (e.g., the 10% with the highest entropy) improve the average Dice score and provide a more reliable estimate of the implant area?**

Chapter 5

Implementation

In the next chapter, we discuss our implementation of hip implant detection and segmentation. This will include data collection for our dataset and its characteristics, dataset preparation and augmentation, deep learning model selection, training and optimization, and finally validation and testing.

5.1 Data

Data is the main building block of any research. It is the element that enters the neural network and it is very well known that if we feed poor data into the neural network we cannot expect a quality output. This is especially critical in the case of medical data, such as X-rays, where even small deviations or artifacts can have a major impact on the accuracy of diagnostic results.

5.1.1 Data collection

Data collection was a crucial step in the process of developing and validating deep learning models, especially in cases of medical diagnostics based on X-ray

image analysis. The quality, quantity and representativeness of our data have a direct impact on the success and accuracy of our final model. In our work, data acquisition was one of the most challenging steps, which needed close collaboration with medical experts and cost significant time.

It was important to explain to the doctors how much data is really needed to obtain reliable and high-quality results. This is even worse because in the field of medicine, it is difficult to obtain a wide range of variability, such as different angles of imaging or different anatomical differences between patients. Not to mention the annotation process, where each image should be carefully labeled by an expert. To ensure this, it costs a lot of money and time.

5.1.2 Dataset

The data used in our research was obtained in collaboration with the hospital in Martin, Slovakia. Thanks to them, we gained access to real X-ray images of patients with hip implants. Unfortunately, the data we obtained contained images of only one type of implant, specifically Pulchra type. This meant that we were unable to work on the possibility of classifying multiple types of implants, because we would need X-ray images of other types of implants as well.

The data we obtained from the hospital chief were also not the most optimal for our purposes. It contained images where two implants were often present at the same time in one image. In addition, the data also contained green lines and annotations with information about the dimensions of the implants. However, these annotations represented visual noise for our purposes, and thus reduced the quality of the images themselves. Therefore, we used a modified dataset that ensured that each implant was on a separate image. The images were also cropped to show only the essential part - the hip implant. At the same time, the green

lines with annotations were also removed. Thanks to this modification, we were able to work with better-quality input data and thus strengthen the accuracy of our neural network. The final dataset then consisted of:

- 94 images of control implants
- 112 images of loose implants

and was prepared to be used in our neural network.

5.2 Data Preparation and Augmentation

Working with the obtained data also played a key role in our research. Therefore, in this section we will focus on data preparation and augmentation, as we needed to ensure the reliability and generalizability of the model.

5.2.1 Pre-processing

Data preparation is not just about getting data, we need to clean and process it properly. We had to ensure consistent dimensions of our images. We chose 256x256 pixels as the input image size. This size is neither too small that we have trouble capturing the details of the implants nor too large that we spend an unreasonably long time training.

In addition, we normalized our data, because machine learning models, as mentioned in the analysis section, are based on specific ranges of values.

Of course, we also tried to minimize unwanted noise using basic noise reduction filters.

5.2.2 Mask generating

Since we did not have segmentation masks at the beginning of the segmentation task to serve as target outputs for training, we had to prepare masks. We decided to create these masks using a neural network. This network generated exactly one mask for each input image, where each pixel was classified as either part of the implant or as the background. In this part, we tried to experiment more and included several techniques for preprocessing the inputs in addition to two different model architectures.

These techniques had a significant impact on the quality of the input data for our models. The main preprocessing techniques we used include Gaussian Blur and adjusting brightness, noise removal, edge detection using Canny Edge Detection and Morphological cleaning. Needless to say, brightness and contrast are of course critical factors in medical images that affect the visibility of anatomical structures and therefore implants too. The combination of these all techniques ensured that we obtained clearer and more accurate implant contours, smoothed and continuous mask areas, and eliminated small artifacts and noise.

Thanks to this, the quality of the created binary masks varied during the experiment, which ultimately had a positive impact on the accuracy of our resulting segmentation model.

Regarding the implementation of neural networks, we implemented two variants of the u-net architecture: the classic U-Net and the residual U-Net. First, we chose the classic u-net architecture for the beginning. This is an easier network to implement and verify whether it will work for us, but this option also comes with a lower computational load and a higher training speed. This network efficiently generated initial masks for us with reasonable accuracy.

Although we generated the initial masks to some extent, we tried to get better results because some images still remained empty or other masks were not perfectly accurate. We noticed a limitation, especially at the edges of the implants. We decided to try another architecture. This was the Residual U-net, which should improve the original classical architecture by using residual blocks. These blocks introduce skip connections within individual layers. This should allow for more efficient gradient transfer and more stable training. Our Residual U-Net achieved more accurate results compared to the classical U-Net, especially when segmenting fine border areas of implants [5]. A comparison of the generated masks can be seen in the figure 5.1.

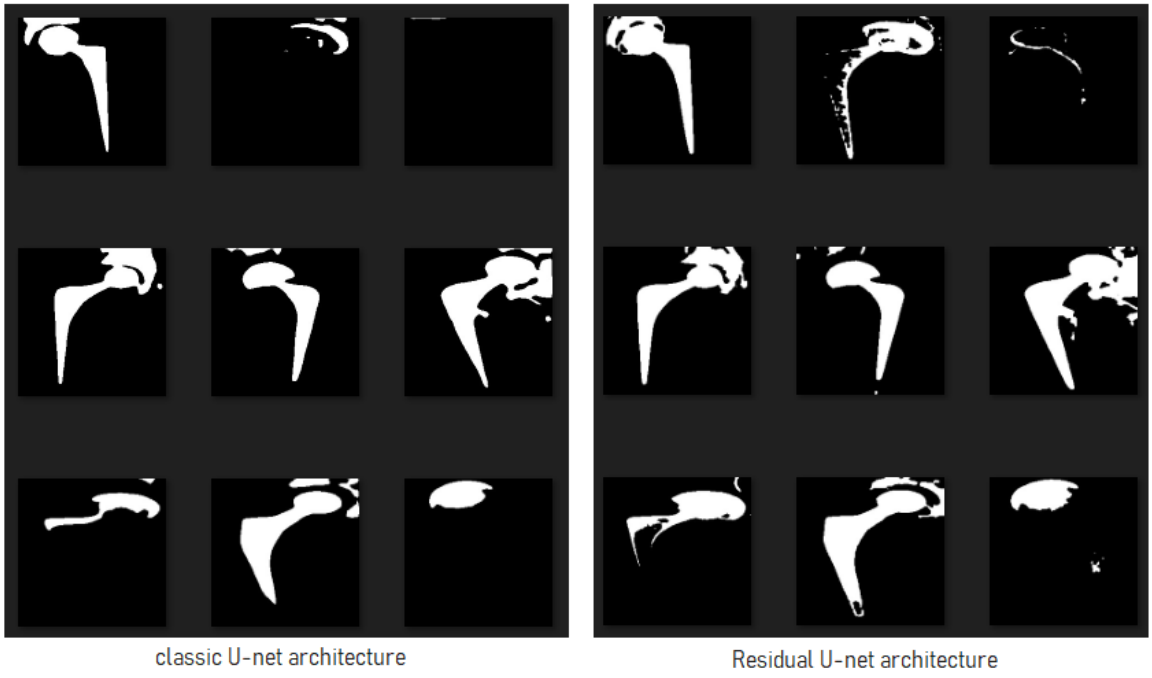


Figure 5.1: Comparison of segmentation results of the classic U-Net (left) and Residual U-Net (right).

After successfully generating the masks, we could use them as input data for the second phase, implant segmentation.

5.2.3 Augmentation

Data augmentation is again an important part of dataset preparation, especially when we have only a limited amount of data available. It gives us a chance to expand our, in our case relatively small dataset by artificially generating variations of existing data. Thanks to this, the model can generalize to new unknown inputs.

We applied several basic augmentation techniques like:

- Rotation - so that the model can recognize implants regardless of their orientation on the images.
- Horizontal flip - so that the model is not sensitive to the right or left orientation of the implants

We applied these augmentation techniques randomly to increase the diversity of the training set. The augmentation was also applied in real time during the training process so that we did not need to store a huge number of artificially generated images.

5.2.4 Data split

The final step in data preparation was to divide the data into three sets:

- Training Set: 70% of the data was used to train the model.
- Validation Set: 15% of the data was used to validate and fine-tune the model.
- Test Set: 15% of the data was used to evaluate the model's performance.

Carefull preprocessing, high-quality mask generation, and diverse augmentation allowed us to create a reliable dataset. While reducing the possibility of overtraining

and enhancing the model's capacity to generalize to new data, this dataset provided a solid foundation for training and evaluating our segmentation model.

5.3 Model Selection

After data preparation, we proceeded to the second key phase of the project - the selection of the neural network architecture for the segmentation of the hip implant. Our objective was to evaluate various families of models, each representing a distinct image processing philosophy, and to validate how their strengths manifest themselves in hip implant X-ray images. That is why we have included four architectures in our work.

For our segmentation implementation, we decided to use the U-Net architecture. Related work also guided us on this path, which is not surprising, because this architecture was designed specifically for medical image segmentation [6]. In addition, we also used Unet++, TransNet and EffNet architectures.

5.3.1 U-Net Implementation

U-Net is a deep convolutional neural network that was designed in 2015, specifically for image segmentation tasks, especially in the field of medicine. Since its introduction, it has become the basis for many applications in medical image analysis. The name it received comes from the characteristic shape of the architecture, which resembles the letter "U" [57].

The choice to include this architecture was mainly driven by its ability to effectively combine global and local image features while maintaining high accuracy and therefore be able to identify and localize relevant structures well even when we have a limited amount of data [6].

The architecture of U-Net can be seen in the figure 5.2 below.

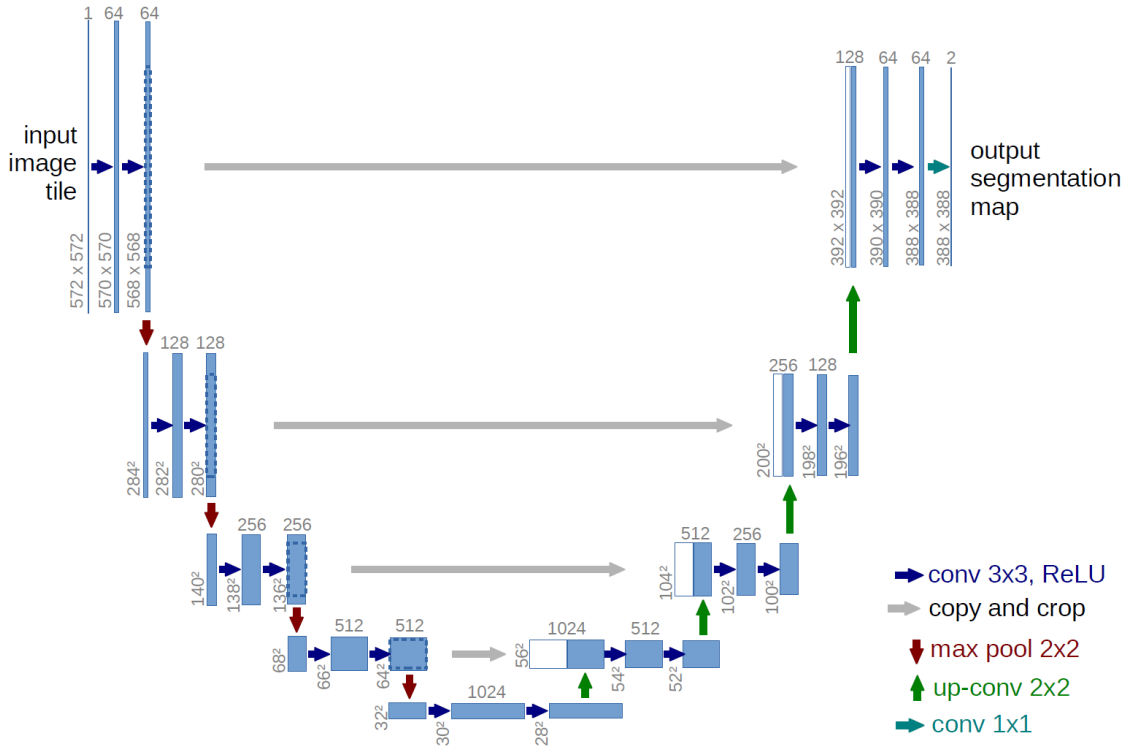


Figure 5.2: U-Net Structure: The architecture consists of a contracting path (encoder, left side) for feature extraction, an expansive path (decoder, right side) for resolution restoration, and skip connections to preserve detailed information [57].

The structure consists of two main parts: a contraction (encoder) path and an expansion (decoder) path [39].

Contraction path (Encoder):

- At each step, two convolutional layers with a small kernel (usually 3x3) with a ReLU activation function are applied. In the architecture figure, this part is represented by blue arrows.
- Then comes the max-pooling (2x2) with a step of 2, which reduces the image resolution by half. Thanks to this, we can extract deeper and more complex

features. This is represented by red arrows.

Expansive path (Decoder):

- Here again, 3x3 convolution blocks with ReLU activation function are applied in each step. In the figure, again blue arrows.
- Then instead of down-sampling with max-pooling, the decoder uses upsampling operations and 2x2 transposed convolutions, which thus doubles the spatial resolutions of the feature maps and restores features that were lost during the encoding phase [57]. This is represented by the green arrows.

There are also two types of connections between the encoder and the decoder:

- **Skip Connections (grey arrows in the figure):** these connecting parts are very important part of U-Net, they take the outputs of the symmetrical part of the encoder and cancatenate them onto their opposing stage in the decoder. Thanks to this, our model preserves fine details even during decoding.
- **Bottleneck:** This connection works as a bridge between the encoder and decoder. Here we down-sample the features, pass them through the convolutional layers and then up-sample the features to get back to the previous resolution before the bottleneck. This allows it to capture the most critical features while maintaining spatial information.

In the end, 1x1 convolution is used. It maps the output of the last decoding layer to desired number of classes and each pixel is classified as part of the segmented object or background.

We have included this model in the comparison with other models mentioned later in the section 6.1. Based on that, we used two types of U-Net models in

different phases of mask generation and segmentation, specifically the classic U-Net architecture and the Residual U-Net architecture. Their detailed implementation in terms of mask generation was presented in section 5.2.2. The implementation of UNet for segmentation purposes is presented in the chapter 5.4.

5.3.2 TransUNet Implementation

Traditional convolutional networks used for segmentation, especially U-Net, are quite effective at modeling local texture features, but suffer from a limited receptive field for large anatomical structures. However, transformer models, on the other hand, are excellent at capturing global context, although they lack fine localization accuracy. In order to have the best of both worlds, we decided to use TransUNet as a second model, which should be more sophisticated, to extend the original U-Net pipeline. This TransUNet model combines the best features: convolutional "stage" encoder-decoder preserves precise spatial information, and an embedded Vision-Transformer encoder models longitudinal dependencies. This results in a much improved edge sharpness while maintaining high Dice scores on medical data [15].

The structure of this model can be seen in the figure below 5.3. It appears to be the following:

1. The Input Image: The X-ray image is first scaled to a size that is divisible by 16. In our work, we used images scaled to 256x256px. The image has three channels (RGB or 3x grayscale copy)

2. The encoder part of U-Net: Conv-blocks (3x3, ReLU) that progressively decrease in resolution. After first block $\rightarrow$ (H/2, W/2), after second block $\rightarrow$ (H/4, W/4), after third block $\rightarrow$ (H/8, W/8), after fourth block $\rightarrow$ (H/16, W/16). By doing so, the output of the last block has C=768 channels and

provides the image's compacted representation. This part is designed to learn the detection of fine edges and local patterns and is represented by the purple blocks in the attached figure.

3. The Attention layer: is responsible for conversion to tokens and Vision-Transformer. The 1/16-th feature map is sliced into 16x16px patch tokens. Each patch is then projected linearly onto a 768-length vector. The vectors are given a sinusoidal position and a class token is added. The sequence passes through 12 transformer layers, each employing a combination of Multi-Head Self-Attention and a multi-layer perceptron (MLP). This layer allows the model to understand the global shape of the implant, specifically the relationships between distant points. It is depicted by the green block in the figure.

4. Translation back to 2-D map: Following the transformer, the sequence is reassembled into a 1/16 grid (H/16 x W/16, C = 768).

5. Decoder/Up-sampling - A 1x1 convolution operation will reduce the channel dimension to 512, thereby activating the decoder. Bilinear up-samplings or transposed convolutions increase the map back to H×W. During each iteration, the maps are concatenated with the corresponding purple maps from the encoder (skip-connections). An application of a last 1x1 filter results in a logit transformation (probability estimation of pixel belonging to an implant). The blue blocks depicted in the figure are the subject of this.[16].

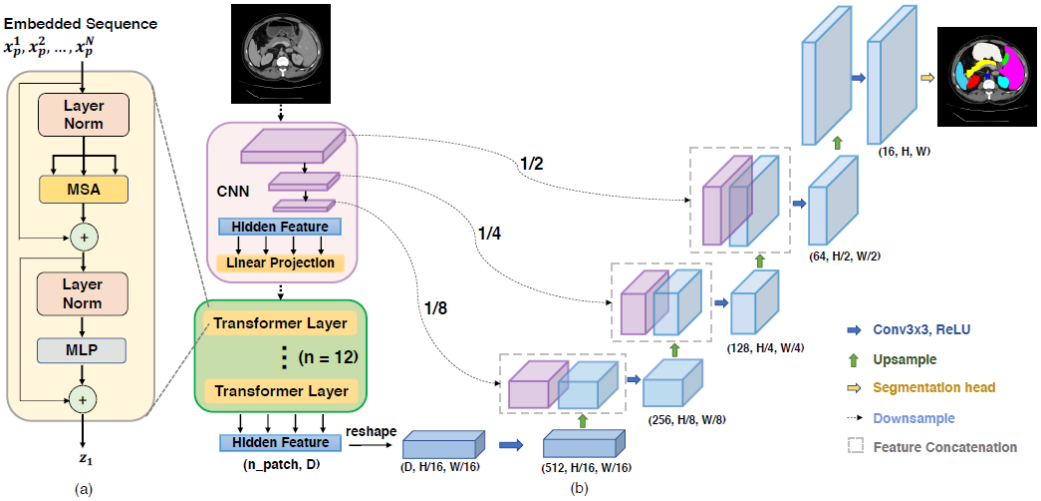


Figure 5.3: TransUNet Architecture [16].

Based on this theoretical foundation, we comprehensively trained the TransUNet implementation on the same dataset and with the identical training procedure as the other models presented in the paper. Summary metrics are available in the evaluation chapter of 6, where TransUNet emerges as one of the four approaches analyzed, alongside EfficientNet-U-Net, UNet++, and conventional U-Net.

5.3.3 UNet++ Implementation

UNet++ (commonly referred to as Nested UNet) emerged from the observation that the classical U-Net architecture, despite employing extensive skip-links, establishes connections between the encoder and decoder maps at varying levels of abstraction. Therefore, the authors conceived the concept of 'filling' the gap between these connections with a series of additional convolutional blocks that progressively reduce the semantic gap. The outcome is a densely populated grid $x^{i,j}$ that can be seen in figure below 5.4 [**zhou2018unet**].

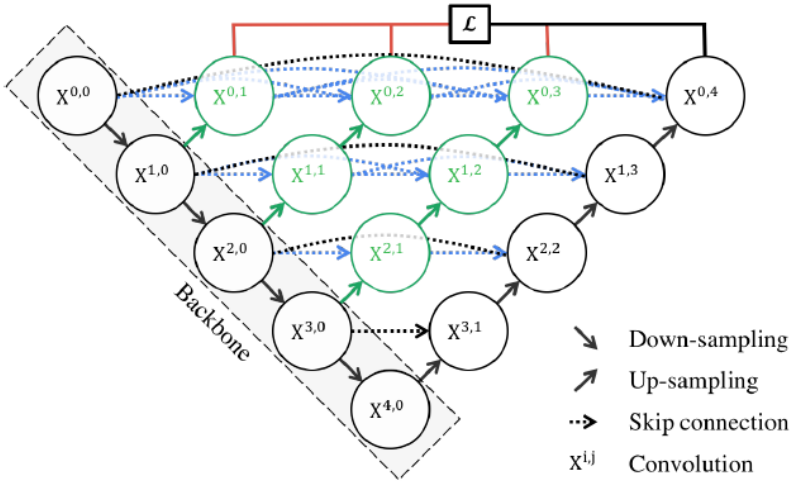


Figure 5.4: Schematic representation of UNet++: newly inserted green ones are representing up-sampling smooth transitions, blue dotted arrows signify classic skip-connections, and red edges denote deep-uppervision branches. Consequently, both low- and high-level features are merged within each node, resulting in the decoder receiving more "cleaner" inputs in terms of content [69].

To make the comparison as fair as possible from an implementation perspective, we used the same ResNet-34 convolution branch as the backbone as was used in the classical U-Net to make the comparison as fair as possible. Also, we trained this model with identical hyper-parameters as the other models: a batch of 16 256x256 images, the Adam optimizer with combined loss of Dice (0.7) + BCE (0.3).

A detailed summary of the results is given in the evaluation chapter 6, where UNet++ appears as the third of the four compared approaches, alongside TransUNet, EfficientNet-U-Net and classical U-Net.

5.3.4 EffNet Implementation

In terms of this work, we decided to implement and test a variant based on EfficientNet alongside the classical U-Net, since the literature has repeatedly shown

that this family of networks produces higher accuracy and noticeably reduced processing needs for the same number of parameters. Additionally, EfficientNet demonstrated a slight improvement in Dice score, by a few percentage points, compared to equivalently sized ResNets, in studies focused on segmenting lung lesions or brain tumors. Therefore, we wanted to see whether the same efficacy would be observed when segmenting hip implants [60][17].

The structure of this variant can be imagined as two cooperating parts:

1. EfficientNet-B0 convolutional encoder: The 256x256 resolution image first undergoes a "stem", where a 3x3 convolution with SiLU activation halves the resolution to 32 channels. Then, four sequential levels of so-called MBConv blocks are applied. Each MBConv block initially expands the channel dimension by an inner 1x1 convolution. Subsequently, it processes the expanded map using a depth-separated 3x3 convolution. Finally, the block compresses the expanded map back to its original channel dimension using a second 1x1 convolution. During this process, a squeeze-and-excite mechanism is used to learn and recalibrate channels based on the global context. Following each block group, the resolution is halved (H/2, H/4, H/8, H/16), and the channel width is progressively increased until it reaches 320.

2. U-Net style decoder: The deepest feature map (320 channels, H/16) is entered into the initial transposed convolution. Afterward, for each resolution enhancement, the corresponding encoder map with the same spatial dimension is concatenated to preserve fine boundary information. This process is alternately repeated until the original 256x256 resolution is reached again. The final 1x1 convolution generates a single-channel logit. After applying the sigmoid function, we obtain a probabilistic map of the implant [21].

In our research, we have successfully implemented this two-part architecture.

Again, the results of our experiments are presented in the evaluation chapter 6, where the model stands next to TransUNet, UNet++ and classical U-Net as the second approach evaluated.

Following implementation, we conducted an evaluation of each architecture presented in section 6.1. Comparative analysis revealed that the U-Net baseline demonstrated the optimal compromise between Dice accuracy, robustness in diverse scenarios, and computational resource consumption. Furthermore, the ensuing epistemic uncertainty, which will be evaluated in the last part of the implementation, is easier to interpret due to its simplicity. Based on these findings, we therefore chose U-Net as the main working model for additional testing and implementation, which we then used to generate hip implant segmentations and perform uncertainty analysis across the entire research.

5.3.5 Segmentation Model Training and Optimization

One of the important parts of our implementation was also hyperparameter tuning. In our hip implant segmentation project using the U-Net architecture, we experimented with several key hyperparameter combinations that affected the performance of our neural network.

The most important hyperparameters that we tuned include:

- **The batch size** that determines the number of images processed by the model before updating the weights. The optimal batch we observed was set to 16, which represented a compromise between stability and training efficiency.
- **Optimizer** The optimization algorithm determines how the network weights are updated during training. We used the Adam optimizer, that should com-

bine the advantages of adaptive learning and moment methods. Adam has demonstrated efficacy in segmentation tasks that require rapid convergence.

- **The learning rate**, the process of modifying the network weights in order to minimize loss following each batch. We initialized model with 2.3337×10^{-3} ; this rate was small enough to prevent gradients from exploding, but large enough to reduce the loss function in the first episodes without the need for a long warm-up.

- **Number of Epochs** The number of epochs determines how many times the model goes through the entire training dataset. We found that 30 epochs was the ideal number based on tracking the validation loss. Additionally, we also implemented Early Stopping, that stopped training when the validation loss stopped improving.

- **Loss function** Regarding the loss function, we tried both Binary Cross-Entropy Loss (BCE) and Dice Loss and their combination. **Binary Cross-entropy Loss** is one of the most commonly used loss functions for binary segmentation. Its mathematical formulation is as follows:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Where:

  - N is the number of pixels in the image
  - y_i is the true pixel value (0 for background, 1 for object)
  - $\hat{y}_i$ is the predicted probability for the pixel

However, its use may suffer from class imbalance. This is possible in X-ray images where the background area is usually much larger than the implant

area, which leads to the model favoring background prediction over background.

Dice Loss, on the other hand, is particularly effective at smoothing out uneven class distributions. It works by measuring the degree of overlap between the predicted mask and the actual mask, and its formulation is:

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + \epsilon}$$

Where:

- N is the number of pixels in the image
- y_i is the true pixel value (0 or 1)
- $\hat{y}_i$ is the predicted probability for the pixel
- ϵ is a constant to prevent division by zero

However, the problem is that stability is lost when the number of positive pixels is small.

In the end, we chose their combination as optimal, because it was motivated by their compensation of strengths and weaknesses. Using BCE, we gain stability and accuracy at the pixel level, where on the other hand, Dice loss takes care of the correct overlap and preservation of the edges of the implants. We used equal weights ($\alpha = 0.5$, $\beta = 0.5$), to maintain a balance between penalizing pixel-level errors and overall mask similarity. The resulting combined loss function is defined as:

$$L = \alpha L_{BCE} + \beta L_{Dice}$$

Where:

- α and β are weights that adjust the importance of each loss function.

5.4 Segmentation of Implants

In this phase, the segmentation of hip implants took place. We chose U-Net for this task again. Although this implementation was based on the classic U-Net architecture, it was modified for better segmentation of anatomical structures.

The difference between U-Net and the original classic version is mainly in the design of individual blocks within the architecture. In the contraction path, instead of simple convolutional layers, we are using DoubleConv blocks, that should ensure more efficient feature extraction and better gradient transfer. In the decoding path, instead of simple upsampling, we use transposed convolutions (ConvTranspose2D) to improve detail reconstruction. The bottleneck layer in the modified U-Net uses DoubleConv blocks, too, so it increases the model's capacity for capturing global properties. Moreover, the skip connections in this new version convey more contextual information by connecting the outputs from DoubleConv blocks rather than a standard convolutional layer [57].

Based on our experiment, the choice of our architectures led to a relatively efficient approach to solving the problem of segmenting hip implants. An appreciable advantage of this choice of architecture was the ability to process even a smaller dataset efficiently and achieve good results, because U-Net maintains a high level of generalization thanks to skip connections.

5.5 AI Uncertainty Implementation

After training the segmentation networks, we added the estimation of epistemic uncertainty to the pipeline.

5.5.1 Epistemic uncertainty method

Three practical requirements were taken into consideration when selecting a method to quantify epistemic uncertainty: it must not alter the architecture of networks that have already been trained, it must be able to operate on the same GPU without causing a visible time penalty, and it must offer a theoretically supported and validated approximation to Bayesian averaging. All three requirements are met by Monte Carlo dropout.

For all architectures, we retained the dropout layers from training and switched them them to the train() mode during inference, but only the dropout layers are actually activated; all others - convolutions, batch-norms or attention blocks - remain in the values they were stored in after training. Consequently, a single frame traverses the network N times. Based on that probabilistic maps we generate:

- average mask $\overline{P}$,
- pixel metrics - variance, standard deviation, IQR and 95% range
- the entropy of the diameter $H(\overline{p})$,, which we chose as the main indicator of epistemic uncertainty after validation.

5.5.2 Metric selection

Comparing pixel measurements was one of our initial actions. In the first row, we displayed the original, ground-truth, and average segmentation; in the second row, we visualized five maps (var, std, entropy, IQR, range95) on typical pictures. We

also calculated the variability of all properly segmented pixels and the variability of incorrect pixels individually for each measure. A decent measure should meet the following criteria when we obtain an estimate of the epistemic uncertainty for each pixel:

- Low uncertainty -> correct pixels (when the segmentation agrees with GT)
- Higher uncertainty -> incorrect pixels (where we hit the wrong one).

The result is a pair of histograms and a number representing the "overlap area," which measures the degree of overlap between the distributions. If we display these two groups as two histograms and they heavily overlap, the metric is unable to distinguish the errors. On the other hand, the small overlap means that uncertainty metric is informative - we can use it to “warn” the user about pixels that are not to be trusted. In section **??**, we have therefore qualitatively shown that entropy best captures uncertainty, particularly around the implant’s edges.

5.5.3 Influence of the number of samples N

In another part, we wanted to demonstrate how rapidly the segmentation accuracy itself and the epistemic uncertainty stabilize as the number of dropout passes increases. For every architecture, the process was always the same. A set of values N=0,10,20,30,40,50 was first defined. Since N=0 serves as a deterministic baseline, the calculated variance and entropy are both zero so the dropout is totally disabled during inference, leaving us with a single mask. After doing as many stochastic runs on the complete test set for each extra N, we recorded the average probability map, binarized it, and calculated the Dice metrics. In order to draw a Dice convergence curve with a growing number of samples at the end, we continually recorded the means and standard deviations during the data gathering process.

In parallel, we calculated an entropy map for a single representative X-ray picture at each N from the test set. In order to visually examine whether and where substantial changes in the uncertainty distribution persisted at increasing sample counts, we kept these maps side by side. We learned a crucial piece of information from the entire experiment: how many passes it takes for the dice to stop changing.

5.5.4 Median, Mean in Segmentation Maps

We next investigated whether the mean of the dropout samples truly represents the most appropriate "consensus" segmentation maps, or whether it would be more accurate to use the median. The reason for this was the mean's statistical characteristic, which states that it is only an ideal estimator when the samples are symmetrically distributed and have variances that are close to normal distribution.

We added a feature to our uncertainty computation method that, rather of using the arithmetic mean, calculated the pixel median over all samples. Next, using the same test set, we made two predictions: one using the mean map and the other using the median map. We calculated the Dice score for each against the ground-truth mask, and we repeated the same process thirty times using a different random number generator seed to make sure the outcome was unaffected by the specific distribution of the dropout masks. We calculated the mean values and standard deviations from the dice pairs that were so produced, and then we used a paired t-test to assess if the difference between the mean and median was statistically significant.

To further illustrate the point, we included visual demonstrations in addition to the numerical comparison. For a chosen image, we showed the entropy maps from

the mean, median, and difference map side by side. The difference map color-coded the pixels where the two methodologies differed.

5.5.5 Error-Rate and Uncertainty

The aim of the following task was to determine whether the model produces the most mistakes in the exact places where it admits a high degree of uncertainty. In order to do this, we created a two-dimensional histogram map, in which the vertical axis displays the percentage of pixels that were incorrectly segmented (error-rate), while the horizontal axis depicts the range of epistemic entropy. The following steps made up the procedure:

- Computation of the error mask and entropy. We used 30 dropout samples to create an entropy map for every test image. A binary error mask was also made in parallel; a pixel is set to 1 if the projected class does not match the ground truth and to 0 otherwise.
- We created $K = 20$ uniform bins out of the whole entropy range $[0, \ln 2]$.
- For each frame, we then selected all pixels lying in a given entropy bin and counted the fraction of them that were erroneous. This resulted in the pair (u,k), where k is the slice-wise error-rate and u is the center of the uncertainty bin.
- The number of records in each cell combination was then calculated. After normalizing the counts to [0,1], we visualized them; a cluster of bins with significant error rates is shown by dark red dots.
- Finally, we took the average of all adjacent error-rates for each entropy bin and plotted it as a red poly-line over the heat-map. The curve visually shows the trend: a rising line means that increasing uncertainty does indeed signal

a higher error rate.

5.5.6 Structural Uncertainty and Structural Error

Then we moved from pixel maps to the structural level, to see how uncertainty behaves when looking at the entire implant in a single image. For each record in the test set, we first generated thirty dropout segmentations. From each mask, we counted the number of pixels belonging to the implant and thus obtained thirty volume estimates $V_1, \ldots, V_{30}$. From these values, we derived the arithmetic mean μ_V and the standard deviation σ_V. Their ratio VVC $= \frac{\sigma_V}{\mu_V}$ is the Volume-Variation-Coefficient - a numerical measure of how much the volume varies across dropout samples, thus quantifying structural (epistemic) uncertainty. In parallel, we calculated the classical Dice score between the average mask $\overline{P}$ and the ground-truth contour for each frame. Since we were interested in the error, we used the value $1-\text{Dice}$ and thus obtained the pair $(\text{VVC}, 1-\text{Dice})$ for each test frame.

We constructed a scatterplot of all pairs: the horizontal axis shows the segmentation error, the vertical axis shows the volume variation. To see the trend, we fitted the points with a linear regression and added the slope statistic, the coefficient of determination R^2 and the p-value.

5.5.7 The Impact of Augmentation

To investigate whether data augmentation during training leads to more reliable - that is, epistemically more certain—segments, we decided to train two identical UNet-34. The only difference was in the training data.

The augmented model processed the X-ray images scaled to 256x256px without any transformations. On the other hand, the augmented model used the same images, but with each batch we applied horizontal mirroring and random rotation

from the interval ±20 with a 50% probability. Again, we kept the hyperparameter settings uniform to ensure a fair comparison. The models were trained until an early stop was triggered with a compound loss of $0.7 \cdot \text{Dice} + 0.3 \cdot \text{BCE}$, identical batch-size = 16 and learning-rate = 2.3337×10^{-3}.

In the testing part, we ran MC-dropout with 30 passes on each model. For each image in the test set, we calculated the average mask, the Dice relative to the ground-truth, and the entropy of the average as a pixel-wise measure of epistemic uncertainty. This generated two paired sets of numerical results.

At the level of the entire dataset, we then calculated the average Dice and average entropy for both models. To compare them, we used a paired t-test to verify whether the differences in Dice and entropy were statistically significant. For visual comparison, we selected one X-ray image and plotted the predictions end entropy maps of both models, with difference map of entropy included.

By using this approach, we may come closer to determining whether augmentation during training actually lowers model uncertainty or if it only modifies the distribution of errors without changing their amount.

5.5.8 Removing the 10% Most Uncertain Pixels

One way we wanted to check whether epistemic uncertainty makes sense to implement in segmentation and therefore whether the calculated epistemic uncertainty really points to places where the model is wrong is to remove the 10% most uncertain pixels. In other words: if we "cut out" those pixels where the network admitted it is most uncertain, will the accuracy on the rest of the image improve?

For each image in the test set, we generated thirty dropout predictions, from

which we generated an average probability map and an entropy map. We sorted the entropy values into a one-dimensional array and chose a threshold T such that exactly 10% of the pixels had an entropy higher than T. This created a binary "validity mask" M, in which the number 1 indicates pixels left for evaluation and 0 indicates pixels with the highest uncertainty. First, we calculated Dice on the entire mask in the classic way. Then we introduced our own function, which ignores pixels with $M = 0$ during the calculation, and we obtained the value $\text{Dice}_{\text{masked}}$. We stored the pair $(\text{Dice}_{\text{full}}, \text{Dice}_{\text{masked}})$ for every image. We computed the mean $\pm$ standard deviation of both measures after processing the complete dataset, and we performed a paired t-test to see if the difference was statistically significant. To demonstrate this method visually, we have selected one representative image. The four-frame visualization shows (i) the original prediction, (ii) the entropy map, (iii) the binary valid mask (90% of the pixels retained), and (iv) the same prediction with a red overlay of the 10% of the pixels removed. The image shows the specific Dice values before and after masking.

Based on this, the method also provides a practical rule for future deployment – the doctor can automatically receive a warning if it is necessary to manually check the parts of the mask that the model has identified as most uncertain.

5.5.9 Segmentation Model based on Uncertainty

The last experiment builds on the observation that high epistemic entropy is concentrated in the areas where the network makes the most errors. Thereby, the model should learn to provide segmentations that are more accurate and cautious in conveying dependability if we can gently punish predictions with high uncertainty during training. For this reason, we introduced a modified loss

$$\mathcal{L} = \alpha \,\text{Dice} + \beta \,\text{BCE} + \lambda \,\overline{H(\overline{p})},$$

where $\overline{H(\overline{p})}$ represents the average entropy of pixels in a given batch and λ controls the strength of the penalty. At $\lambda = 0$ we return to the original combination of Dice + BCE; increasing λ forces the network to prefer solutions with lower epistemic uncertainty.

In each training step, we perform K=8 dropout passes. From their average, we calculate Dice and BCE. Then we ran the training five times, each time with a different value $\lambda \in \{0, 0.05, 0.1, 0.2, 0.5\}$. All other settings – batch 16, Adam, learning rate, 30 epochs but with early stop, identical train/val split – remained unchanged to make the comparison fair. After each epoch, we store: average training loss, validation Dice, average entropy and epistemic variance. After all runs were completed, we plotted the validation Dice curve for five different.

This Entropic Penalization Training allows us to experimentally verify whether epistemic uncertainty can be used not only as a descriptive but also as a regularization tool.

After the implementation was completed, we were able to compare and evaluate our results. This can be found in the Evaluation chapter 6.

Chapter 6

Evaluation

6.1 Segmentation models comparison

The base model in our work is the classical U-Net, that has long been considered as the "gold standard" of medical segmentation, because of its efficient skip-connections and its ability to preserve the borders of structures in detail [57]. In order to verify the benefit of denser encoder-decoder bridging, we implementet UNet++, that uses nested decoding branches to reduce the semantic gap between coarser and fine features and usually accelerates the convergence [69]. Since implant segmentation can also benefit from a global context, we added TransUNet. It is a hybrid network that combines a convolutional encoder with a Vision-Transfromer block that models long-term dependencies in the picture and enhance shape recognition of components [16]. The last candidate is EfficientNet-U-Net, or EffNet for short, which makes use of lightweight MBConv blocks with a squeeze-and-excite technique and dynamic width, depth, and resolution scaling. The latter promises decreased VRAM consumption and enhanced precision in the number of parameters [21].

To provide a fair comparison, all models were trained using identical 256 × 256 px pictures, the same loss (0.7 Dice + 0.3 BCE), the same batch size (16), Adam optimizer and the same learning rate (2.3337e-3). After five training epochs, there were clear differences in both the learning dynamics and the ultimate accuracy of the individual architectures.

The UNet baseline has surpassed the Dice threshold of 0.67 in the third epoch and achieved a value of 0.8032 in the fifth epoch. The validation error curve declined steadily - from 8.17 to 0.174 - indicating that the classical U-Net can adapt quickly on X-ray images and does not overfit.

EffUNet started with slightly lower accuracy than the baseline, but after an initial 'stumbling' in epoch 2 (Dice 0.002), the network quickly rehabilitated itself, reducing the val loss by a jump to 0.218 and finishing at Dice 0.7879. Although the final outcome was slightly inferior by approximately two percentage points, it is noteworthy that the model has only 5.2 million parameters, which is one-sixth the size of the U-Net. Consequently, it holds significant relevance in VRAM-constrained environments.

It took three epochs for UNet++ to 'clean up' the nested skip links: it passed the Dice threshold of 0.56 in epoch 3 and caught up to EffUNet (0.7882) with each successive iteration. Although UNet++ has one-fifth fewer parameters compared to the baseline model, its implementation of dynamic decoding branches necessitated a greater number of iterations before the network achieved a stable state and optimized its weight parameters to higher precision.

TransUNet, on the other hand, confirmed that the transformer component is sensitive to dataset size. For the first three epochs, it fluctuated just above zero annd only in the fifth epoch did it reach Dice 0.2633. Thus, despite 86.9 million parameters, it failed to achieve even half the accuracy of the other networks. The

extensive weighting capacity required by the ViT block remains under-trained in our setting.

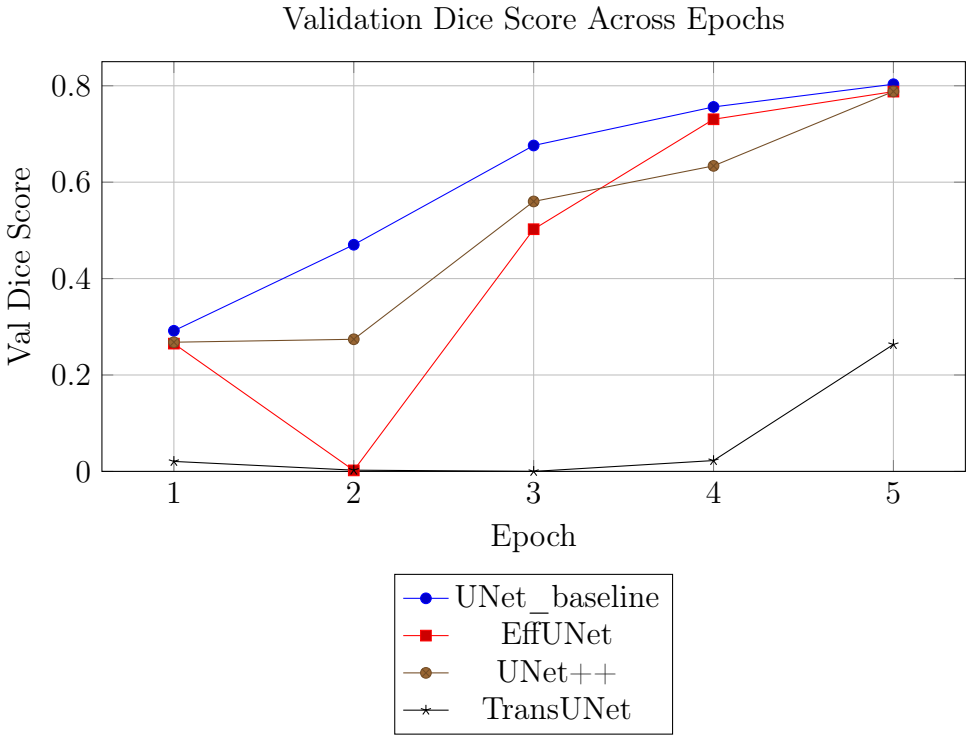


Figure 6.1: Validation Dice scores during training of all models

Model	**Dice**	**Param [M]**	**Epochs**
UNet_baseline	0.802	31.04	5
EffNet	0.788	5.2	5
UNet++	0.788	26.08	5
TransUNet	0.263	86.93	5

Table 6.1: Comparison of segmentation models after 5 trained epochs

The summary table of results that can be seen in Table 6.1 confirms the trade-offs: the UNet baseline is the most accurate, but also the largest of the practically usable models. EffUNet delivers nearly the same Dice with a significantly smaller memory footprint, and UNet++ settles in the middle - with a slightly smaller model

and nearly identical accuracy. The validation curves (Fig. 6.1) demonstrate that all three convolutional architectures converge to a comparable level of 0.78-0.80, although at varying rates. In contrast, TransUNet exhibits a notably lower graph performance. Its potential could only be unlocked by significantly expanding the training dataset or by transfer-learning from a larger domain-related dataset.

Epoch	Train Loss	Val Loss	Val Dice
	UNet_baseline		
1	0.4735	8.1704	0.2917
2	0.3297	1.0675	0.4702
3	0.2658	0.3697	0.6761
4	0.2219	0.2523	0.7561
5	0.1928	0.1740	0.8032
	EffUNet		
1	0.4998	1.0328	0.2651
2	0.3545	0.5548	0.0021
3	0.2973	0.3818	0.5024
4	0.2479	0.2910	0.7305
5	0.2158	0.2183	0.7879
	UNet++		
1	0.5140	25062.4961	0.2680
2	0.2582	23.6986	0.2741
3	0.1702	0.6128	0.5600
4	0.1574	0.9452	0.6338
5	0.1397	0.1553	0.7882
	TransUNet		
1	0.7591	0.6378	0.0207
2	1.0264	0.6828	0.0026
3	0.7454	0.6971	0.0000
4	0.6883	0.6606	0.0225
5	0.6520	0.6297	0.2633

Table 6.2: Training and validation results of all used models after each epoch

In our experiment, the classic U-Net demonstrated superior performance, achieving the highest Dice score with a relatively simple architecture and a reasonable

number of parameters. Therefore, we decided to use it further in our work for both segmentation and epistemic uncertainty determination using Monte Carlo dropout.

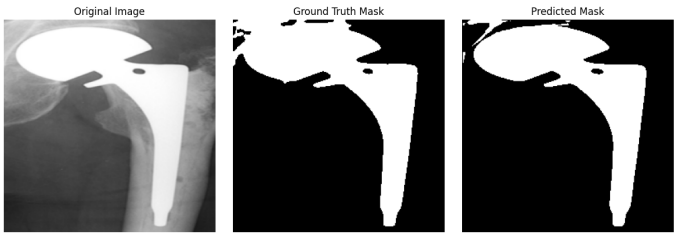


Figure 6.2: Segmentation used UNet baseline

6.2 AI Uncertainty Evaluation

The conventional U-Net proved to be the best design for our data following a series of comparison studies. It obtained an average Dice score of 0.802 on the validation set after just the sixth epoch. The Dice score doesn't reveal anything about the reliability of a certain mask, although offering a rapid indication of segmentation accuracy. However, in order to assess the risk of improper implant size or resection line placement in real clinical practice, the surgeon must know where and to what degree the model may fail. As a result, in the next section of the study, we will examine epistemic uncertainty in addition to traditional accuracy measurements.

The following sections will first demonstrate the decomposition of epistemic uncertainty at the pixel and implant levels, and then confirm its usefulness in practice. For example, we will examine whether eliminating the 10% most uncertain pixels results in a notable increase in Dice or whether entropic penalty in the loss function can lower uncertainty without sacrificing accuracy.

6.2.1 Metrics Selection

We started out focusing on visual examination of their "masks"—images in which each pixel is colored based on the value of variance, standard deviation, entropy, IQR, or 95% range - in order to evaluate the validity of different pixel-wise uncertainty measures. We can see this in the figure 6.3 below. Each metric emphasizes the edges of the implant, as we expected. It is also based on how a radiologist would approach it. The place where multiple doctors who were given the task of marking the implant would disagree the most would be at their edges. However, entropy shows exactly those places where the model fluctuates most between 0 and 1 – typically at the boundaries of the implant, but also in areas with repeated artifacts (screws, metal shadow). Compared to variance, entropy has two advantages:

(1) it is a reliable measure of the uncertainty of binary classification (a pixel either belonged to the implant or not)

(2) its range is bound to [0, ln 2], so it does not require additional scaling or removal of extreme values (as with variance)

From a statistical point of view, we know that variance should be used only when the data distribution is unimodal and resembles a normal distribution.

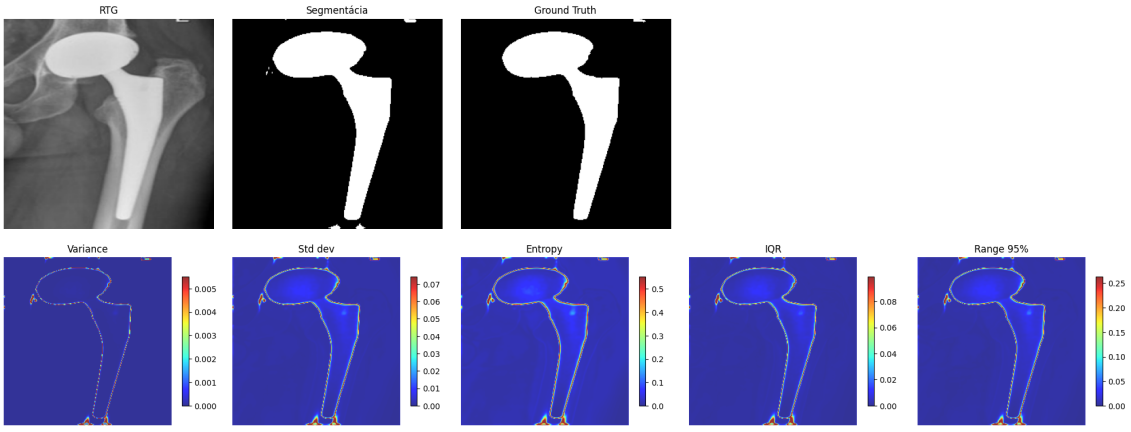


Figure 6.3: Comparison of uncertainty maps based on the used metrics

Plotting a histogram of pixel-wise "overlap" for each case—a measure that counts the number of pixels with an uncertainty value in both tested groups and then overlaps these two distributions—also allowed us to quantify the metrics selection. If we have a huge overlap, something is going on, the metric is wrong and cannot distinguish errors. On the contrary, a small overlap = uncertainty metric is informative – we can use it to "warn" the user about pixels that should not be trusted. As observed in figure 6.4 and table in 6.2.1, the overlap is quite high for the variance and IQR variants (about 0.63 and 0.40), indicating that there is little difference between their core (the majority of pixels) and that it is difficult to distinguish between really uncertain areas.

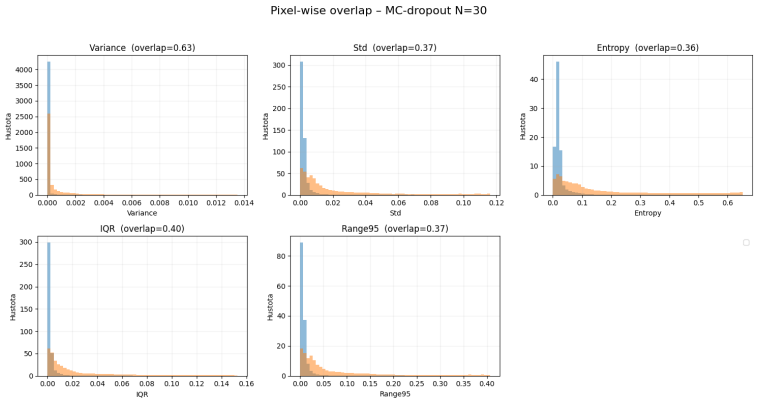


Figure 6.4: Comparison of the overlap area of individual metrics

Metric	**Value**
Variance	0.628
Standard Deviation (Std)	0.372
Entropy	0.360
Interquartile Range (IQR)	0.397
95% Range (Range95)	0.371

Table 6.3: Uncertainty metrics in the overlap area (0 = perfect separation, 1 = complete overlap)

Based on the above, we chose to use entropy. This is particularly relevant as entropy is a common method of assessing epistemic uncertainty in MC-dropout, according to the literature that is currently accessible.

Another question was how many transitions to choose within the dropout. We want to ensure a compromise between dice score and time. The figure **??** illustrates how the number of Monte-Carlo samples N affects the average Dice score on the test set. The baseline score without dropout (N=0), which approaches 0.61, is displayed by the green dashed line. Dice instantly climbs to values of about 0.80 already at N=5 when we enable dropout and progressively increase the number of random runs through the model. From then, it grows just little. The curve flattens out after around 20 samples, and raising N further only slightly improves the situation but greatly raises the computing requirements. Since we already have a virtually steady score in this area while keeping a respectable computational component, we decided to choose N=30 as a working compromise.

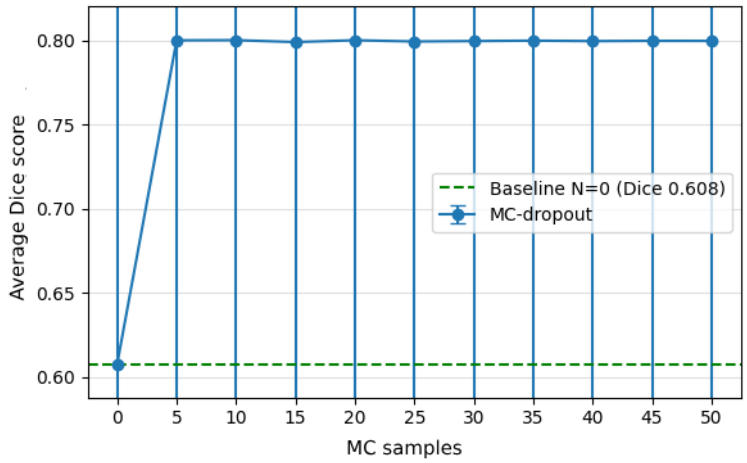


Figure 6.5: Change of Dice score with different MC samples

In addition, we focused on comparing two basic ways of computing Monte Carlo dropout probabilities into the resulting prediction and subsequently into the en-

tropy map – using the mean and the median. The main motive for this comparison was the fact that the mean is optimal if the samples are symmetrically distributed, but is sensitive to outliers that can arise from uncertain pixels. The median, on the other hand, may more accurately represent the average forecast without being skewed by one or two extremely diverse samples and is more resilient to such outliers.

In the figure 6.6 we see different entropy maps for one test slice: on the left the entropy multiplied by the mean p, in the middle by the median p and on the right their difference (median – mean). At first look, the maps appear to be rather similar overall, indicating that our MC-dropout samples exhibit strict multimodality. Nonetheless, the difference map indicates that the median occasionally reduces the implant's most extreme entropy levels.

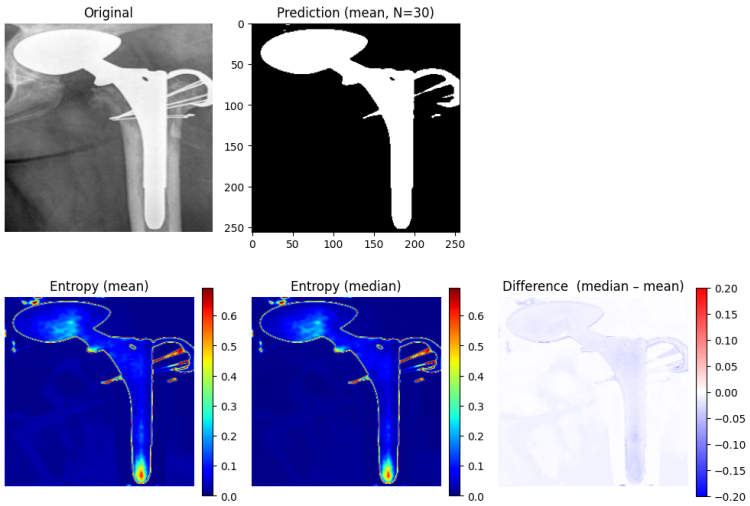


Figure 6.6: Comparing uncertainty maps using median or mean

The average Dice score for the whole test set for both methods is shown in the bar graph in figure 6.7. The numbers are nearly the same:
Mean Dice: 0.7150 ± 0.0003

Median Dice: 0.7149 ± 0.0003

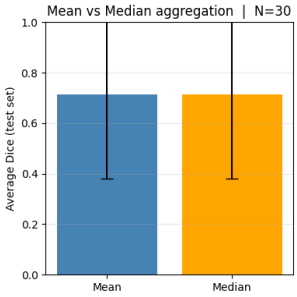


Figure 6.7: Bar plot comparing average Dice scores using median or mean

Finally according to the t-test, we proved that the difference is not significant. This demonstrates that the mean is still more than adequate for our data and that the median does not need to be used.

Another experiment we decided to try was to purposefully compare two models that differed only in whether rotational and mirror augmentations (flip + rotations) were applied during training or not. The main goal was verifying whether such trainig would improve generalization - which means higher average Dice score on unseen data and reduce the epistemic uncertainty of the model. The results are summarized in the table 6.2.1 below:

Metric	**No Augmentation**	**With Augmentation**
Dice Score	0.7178 ± 0.3374	0.7144 ± 0.3348
Entropy	0.0931 ± 0.0136	0.0530 ± 0.0118
Paired t-test p-values		
Dice Score	$p = 2.99 \times 10^{-1}$	
Entropy	$p = 3.38 \times 10^{-23}$	

Table 6.4: Comparison of Dice score and entropy with and without augmentation in model training

From this we can clearly see that training augmentation by rotational flipping and rotation did not bring a statistically significant increase in the average Dice score, on the other hand it significantly reduced the average epistemic entropy. Although we did not manage to significantly improve the accuracy of segmentation, the augmentation helped the model to be more consistent – it hesitated less in its predictions.

6.2.2 The Relationship between Uncertainty and Error

Once we had defined the options for how we would deal with epistemic uncertainty. We could compare how well it correlates with the error rate.

We investigate if our measure of epistemic uncertainty actually corresponds with the mistake rate using the reliability heat-map shown in figure 6.8. The prediction entropy value, which we split into ten equal "bins," is plotted on the horizontal axis. The error rate, or the fractional proportion of pixels that the model categorized wrongly, is plotted on the vertical axis. The number of points, or slice-pixels, in each bin is color-coded. The average error rate for each bin-cross-section is displayed by the red curve.

We see that the average mistake rate rises with increasing uncertainty: errors for low H range from around 5 to 10%, whereas errors for H beyond 0.6 bits reach up to 50%. A statistically significant medium-strong monotonic relationship between uncertainty and real error rate is confirmed by Spearman's correlation.

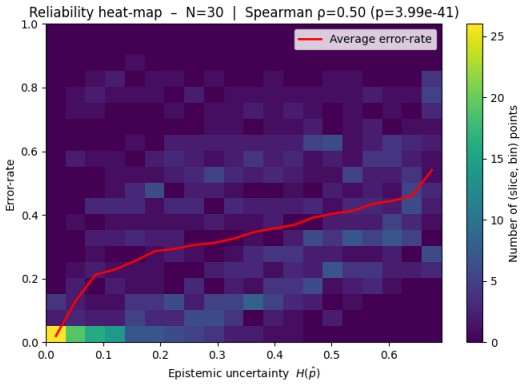


Figure 6.8: Reliability heat-map: The relationship between epistemic uncertainty and incorrect pixels

This finding is valid in sense that our approach is able to pinpoint trouble spots effectively: pixels with a greater prediction entropy are actually more likely to be segregated improperly. This allows us to concentrate future human evaluation or attention on the areas of the image where entropy is most noticeable, which is precisely what reliability analysis aims to do.

We also compare the relationship between volumetric variation (VVC) and segmentation error measured as 1-Dice. In the figure 6.9, each dot represents one case from the test set, with the x-axis showing the decrease in accuracy (1-Dice) and the y-axis the VVC value calculated using MC-dropout. The linear regression indicates a slightly positive trend, which confirms a moderately strong Spearman correlation (p=0.043), i.e. cases with higher variability are more likely to have lower segmentation accuracy.

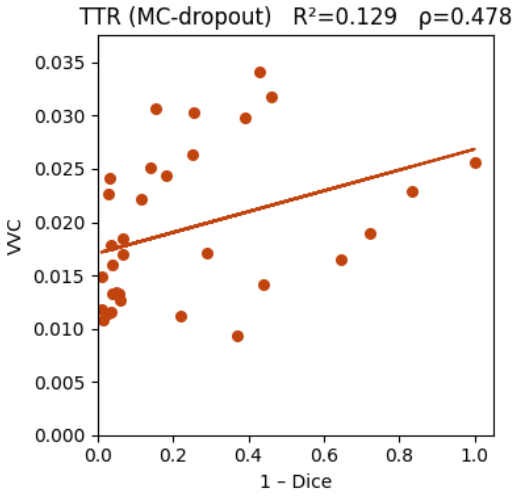


Figure 6.9: The relationship between volumentric variation and segmentation error

6.2.3 Uncertainty Use in Segmentation

In this section, we have already made the basic decisions. Namely, using Monte Carlo dropout, with number of passes = 30 and entropy as the metric. However, we wanted to further explore what advantages the use of epistemic uncertainty can have. Specifically, we focused on practical verification of whether epistemic uncertainty really corresponds to segmentation errors, and whether its use can improve the resulting Dice score.

The ideal tool to show this was the removal of the most uncertain pixels.For each image from the test set, we calculated a prediction (binary mask) and its pixel-wise prediction entropy. We then removed 10% of the pixels with the highest entropy and considered the remaining 90% of the pixels as valid for evaluation. The objective was to demonstrate that eliminating these most confusing regions will enhance the Dice score for the remaining segmentation, which will demostrate the use of uncertainty as a warning sign for dangerous areas.

An example where the original (full-image) dice has a very good value of 0.929

may be found in the figure 6.10. The pixels that we "crossed out" according to entropy are shown by the red ones in the lower right picture. Dice increased to 0.948 after filtering them, indicating that the most ambiguous pixels do, in fact, belong to the edges with potential error and that eliminating them will lower the percentage of incorrect classifications in the remaining mask.

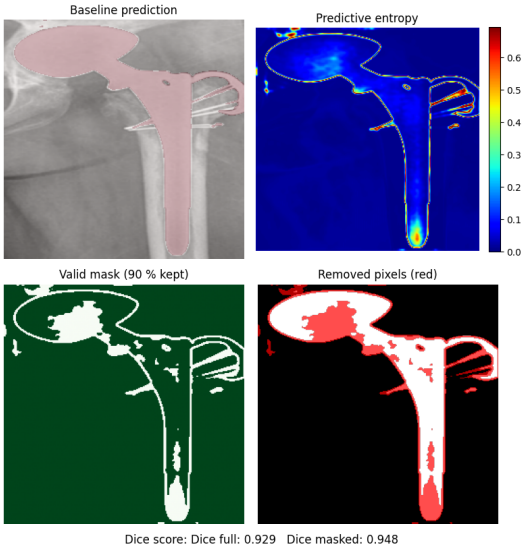


Figure 6.10: First example of Dice before and after removing the 10% most uncertain pixels

We also graphically display the change in dice score on the graph in figure 6.11.

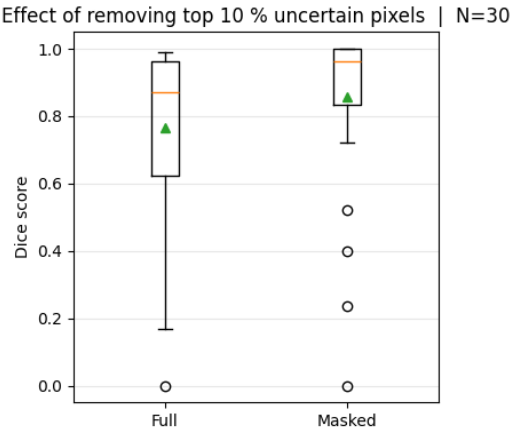


Figure 6.11: Change in dice score after removing the most uncertain pixels

The next figure 6.12 shows a different case: the original Dice is weak - only 0.628, because the segmentation contained quite a lot of errors. The Dice increased to 0.865 once the 10% of pixels that were the most doubtful were eliminated, which is a notable improvement. This demonstrates how entropic uncertainty aids in both "fine-tuning" already-good segmentations and partially "drawing out" possible faults in the inferior ones.

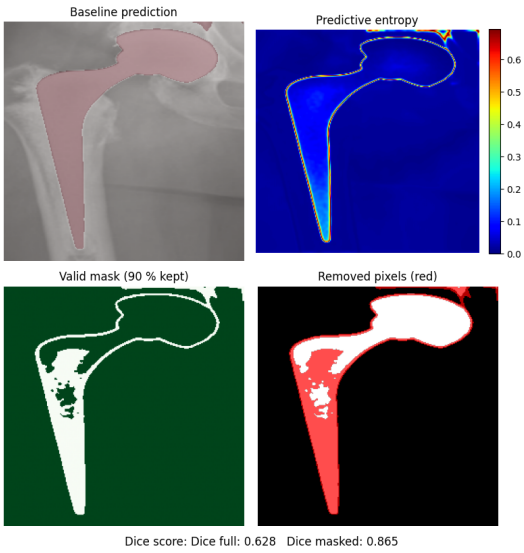


Figure 6.12: Second example of Dice before and after removing the 10% most uncertain pixels

In both instances, we have effectively demonstrated that epistemic uncertainty can not only function as a tool that identifies potential errors, but also as a practical filter that enhances the reliability of the final score on "safe" pixels.

At the end of this work, we moved from a pure MC-dropout model to a model that penalizes its own uncertainty in the form of prediction entropy during training. The goal was to find a compromise between precision (Dice score) and confidence (low entropy), so we could achieve the highest Dice with the lowest average entropy.

First we had to find out what λ should be used. We can see the results in the figure 6.13. To summarize the key findings from the training protocols for different values of the penalty constant λ: $\lambda = 0.05$ had the most balanced behavior: the model produced a val–Dice that was almost identical to the baseline ($\lambda = 0$), but it had a much lower entropy, or a better calibration of its uncertainty. Severe penalties $\lambda \geq 0.02$ decrease the ultimate accuracy while further reducing entropy.

On the other hand, when there is no penalty ($\lambda = 0$), the model "overshoots" in areas of uncertainty, lowering the accuracy of uncertainty indicators. This results in the greatest Dice.

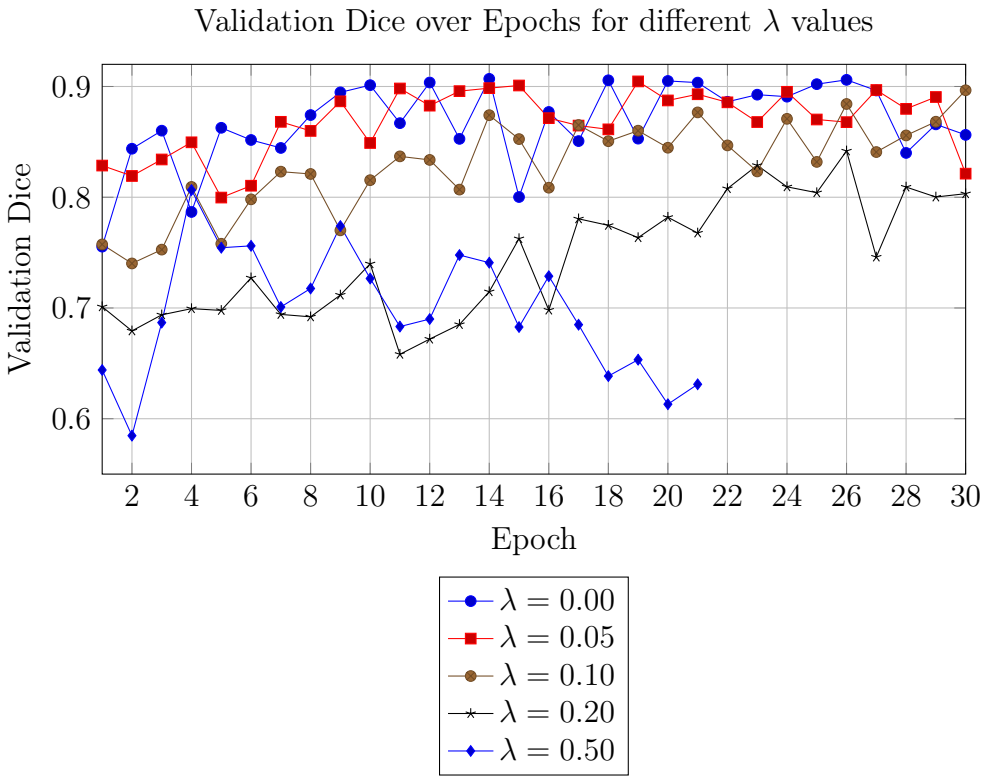


Figure 6.13: Validation Dice score across epochs for different λ values.

Our final comparison of this model with the baseline model on a specific implant can be seen in figure 6.14. The clean X-ray picture (without the segmentation mask) is displayed in the first column. The second column redraws the baseline segmentation edges with a green line and the ground truth edges with a yellow line. It should be noted that the baseline model partially undersegments the prosthesis edge. . The segmentation edges of our $\lambda = 0.05$ model (blue line) and the identical GT edges (yellow) are displayed in the third column. The model matches more closely to the hand annotation in this case. In the second row we see a map of

the baseline model's predictive entropy, the middle image shows the entropy after adding the penalty term. The confidence colors are significantly shifted towards blue. Lastly, the final picture subtracts the baseline entropy from the entropy of the penalized model.

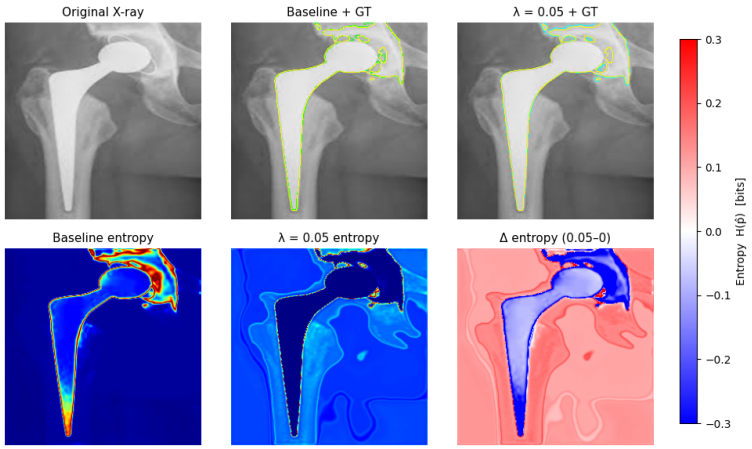


Figure 6.14: Prediction with our final model with $\lambda = 0.05$ on a random X-ray image.

Finally, here we have a figure 6.15 of a qualitative comparison of baseline segmentation and TTD segmentation with epistemic uncertainty. The original X-ray picture is displayed in the upper left, followed by segmentation prediction maps over ground truth. The predictive entropy map for both the baseline model and our TTD model is displayed in the lower panel. The baseline entropy is subtracted from the TTD model in the last panel. Thanks to Monte Carlo Dropout Segmentation (TTD), we can not only improve the sharpness and fidelity of the prosthesis boundaries (bottom right panel), but also obtain an accurate visualization of the uncertainty (bottom left panel). This approach greatly improves interpretability and enables practical deployment when the doctor is searching for trouble spots.

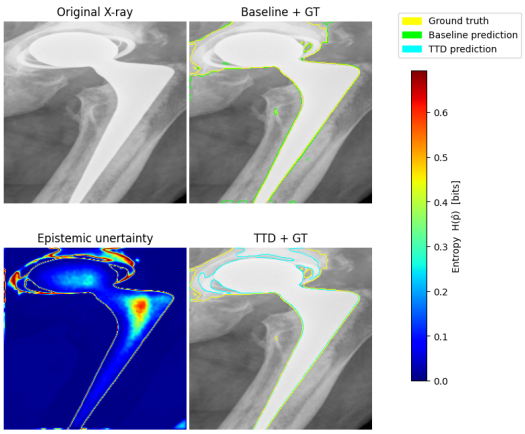


Figure 6.15: Qualitative comparison of baseline segmentation and TTD segmentation with epistemic uncertainty

We suggest this implementation alternative as the final model for segmentation with explicit account of epistemic uncertainty because it achieves an optimal compromise by utilizing a penalizing term with $\lambda = 0.05$.

Chapter 7

Conclusion

We believe that our work makes several significant contributions to the field of medical segmentation and radiological image processing. Our model provides a quantifiable and visualized map overview of the reliability of each prediction. This allows clinicians to quickly identify areas where the model loses confidence and where additional checking or manual correction may be needed. In contrast to purely deterministic segmentation networks, we have developed a practical tool that seamlessly integrates high accuracy with interpretability and transparency of decision-making processes.

In clinical practice, this solution facilitates automated preliminary assessment of implant extent and shape, thereby reducing the workload of radiologists and orthopedists. It also enhances the consistency of measurements, such as assessing wear or potential loosening. The uncertainty mapping functions as an integrated 'alarm' that alerts the operator when the result necessitates specialized attention.

7.1 Summary

In this study, we evaluated the hip implant segmentation using radiographic pictures, focusing on measuring epistemic uncertainty. We first compared four different architectures – the classic U-Net, EfficientNet-U-Net, UNet++ and TransUNet – and found that although transformer-based models bring interesting global features, in our case the best combination of accuracy and computational efficiency was achieved by the baseline U-Net. Next, we experimented with different aggregation techniques (mean vs. median), binning types in the creation of reliability heat-maps, the effect of training augmentation, and Monte Carlo dropout for uncertainty estimation. Lastly, we tuned the hyperparameter λ and created a penalty loss function using entropy. The findings show that: The stability of segmentation performance with increasing number of MC samples stabilized at Dice = 0.80 after approximately N = 30. Entropy proved to be a more reliable indicator of error areas than variance, IQR or 95% range, and it also brought the smallest pixel-wise overlap with errors. Median aggregation instead of mean in MC-dropout did not bring statistically significant improvement of segmentation, which we confirmed by paired t-test. Training augmentation reduced the average entropy of segmentations, although the Dice score itself remained similar. Entropy penalty ($\lambda = 0.05$) provided the most balanced compromise between high accuracy and reduced uncertainty.

To answer the research questions we set out in the chapter 4:

- **RQ1: How can a U-Net convolutional network be designed and trained to reliably segment a hip implant in standard radiographic projections?** Our comparison showed that although advanced architectures such as TransUNet and UNet++ bring advantages in terms of capturing global relationships, in our data and with a given volume of annotations, the

best results were achieved by the classic U-Net due to its simplicity, fewer parameters, and faster convergence.

- **RQ2: Which epistemic uncertainty metric (variance, entropy, IQR, 95%-range) best correlates with pixel-wise segmentation error in Monte-Carlo dropout?** Although we tested multiple statistical measures (variance, standard deviation, IQR, 95% range), entropy provided the smallest pixel-wise overlap with real errors.

- **RQ3: How does the addition of an entropy-based penalization term affect the segmentation accuracy (Dice) and the level of predictive uncertainty relative to the baseline model without penalization?** We significantly decreased the average entropy of the segmentations by adding a penalty to our model implementation. An average entropy of 0.35 bits and val-Dice of 0.90 are obtained by the model with $\lambda = 0.05$.

- **RQ4: Can removing the most uncertain pixels (e.g., the 10% with the highest entropy) improve the average Dice score and provide a more reliable estimate of the implant area?** We saw an increase in Dice by "removing" (mapping) the 10% of pixels with the greatest entropy from each segmentation result. The most uncertain regions are where segmentation mistakes are most likely to occur when the Dice score increases.

7.2 Future Work

To summarize it. There is still a lot of work to be done in the future. It would be appropriate to expand the dataset to include additional implant types and shapes so that the model not only segments but also recognizes specific manufacturing versions and component variants. At the same time, we expect that the increase in

the number of images and the expansion to include new implants will contribute to more robust training and allow for the simultaneous solving of segmentation and classification tasks in one integrated framework.Regarding uncertainty, it is possible to go in the direction of aleatory uncertainty, which would again have some impact on segmentation. Finally, we could combine both methods and achieve even better information about how confident the model is in segmentation.

Literatúra

[1] URL: https://survicate.com/customer-satisfaction/importance-customer-satisfaction/.

[2] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information fusion* 76 (2021), pp. 243–297.

[3] Rolf Adams and Leanne Bischof. "Seeded region growing". In: *IEEE Transactions on pattern analysis and machine intelligence* 16.6 (1994), pp. 641–647.

[4] Mohammad Alamleh et al. "Uncertainty Estimation for Deep Medical Image Segmentation". In: (2020).

[5] Md. Zahangir Alom et al. "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation". In: *CoRR* abs/1802.06955 (2018). arXiv: 1802.06955. URL: http://arxiv.org/abs/1802.06955.

[6] Reza Azad et al. "Medical Image Segmentation Review: The Success of U-Net". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 10076–10095. DOI: 10.1109/TPAMI.2024.3435571.

[7] Fuqin Bao et al. "Application of adaptive threshold image segmentation algorithm in orthopedic CT imaging". In: *Journal of Medical Imaging and Health Informatics* 9.8 (2019), pp. 1736–1740.

[8] Thomas W Bauer and Jean Schils. "The pathology of total joint arthroplasty: II. Mechanisms of implant failure". In: *Skeletal radiology* 28 (1999), pp. 483–497.

[9] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. "The need for uncertainty quantification in machine-assisted medical decision making". In: *Nature Machine Intelligence* 1.1 (2019), pp. 20–23.

[10] Daniel Bell and Candace Moore. *Noise reduction*. Dec. 2019. DOI: 10.53347/rid-72577. URL: http://dx.doi.org/10.53347/rID-72577.

[11] J Berhouet, P Garaud, and L Favard. "Influence of glenoid component design and humeral component retroversion on internal and external rotation in reverse shoulder arthroplasty: a cadaver study". In: *Orthopaedics & Traumatology: Surgery & Research* 99.8 (2013), pp. 887–894.

[12] Yuri Y Boykov and M-P Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images". In: *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*. Vol. 1. IEEE. 2001, pp. 105–112.

[13] Jerrold T. Bushberg. *The Essential Physics of Medical Imaging*. Lippincott Williams & Wilkins, 2002.

[14] John Canny. "A computational approach to edge detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[15] Jieneng Chen et al. "TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers". In: *Medical Image Analysis* 97 (2024), p. 103280.

[16] Jieneng Chen et al. “Transunet: Transformers make strong encoders for medical image segmentation”. In: *arXiv preprint arXiv:2102.04306* (2021).

[17] Sachikanta Dash et al. “Enhancing Lung Cancer Diagnosis through CT Scan Image Analysis using Mask-EffNet”. In: *Engineering Access* 11.1 (2025), pp. 92–107.

[18] “Deep learning-based dental implant recognition using synthetic X-ray images”. In: *Medical & Biological Engineering & Computing* 60.10 (2022), pp. 2951–2968. DOI: 10.1007/s11517-022-02642-9.

[19] Tribikram Dhar et al. “Challenges of Deep Learning in Medical Image Analysis -Improving Explainability and Trust”. In: *IEEE Transactions on Technology and Society* PP (Mar. 2023), pp. 1–1. DOI: 10.1109/TTS.2023.3234203.

[20] Andre Esteva et al. “A guide to deep learning in healthcare”. In: *Nature medicine* 25.1 (2019), pp. 24–29.

[21] Ido Freeman, Lutz Roese-Koerner, and Anton Kummert. “Effnet: An efficient structure for convolutional neural networks”. In: *2018 25th ieee international conference on image processing (icip)*. IEEE. 2018, pp. 6–10.

[22] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[23] Jakob Gawlikowski et al. “A survey of uncertainty in deep neural networks”. In: *Artificial Intelligence Review* 56.Suppl 1 (2023), pp. 1513–1589.

[24] Manivasagam Geetha et al. “Ti based biomaterials, the ultimate choice for orthopaedic implants–a review”. In: *Progress in materials science* 54.3 (2009), pp. 397–425.

[25] Zibo Gong et al. “Automated identification of hip arthroplasty implants using artificial intelligence”. In: *Scientific Reports* 12.1 (2022), p. 12179.

[26] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing*. 3rd ed. Upper Saddle River, NJ: Pearson, Aug. 2007.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[28] Liyao Guo et al. "On the design evolution of hip implants: A review". In: *Materials & Design* 216 (2022), p. 110552. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2022.110552. URL: https://www.sciencedirect.com/science/article/pii/S0264127522001733.

[29] J.R. Haaga et al. *Computed Tomography & Magnetic Resonance Imaging Of The Whole Body E-Book*. Elsevier Health Sciences, 2008. ISBN: 9780323076210. URL: https://books.google.sk/books?id=ii3kS2ih8JsC.

[30] Anna Hadamus et al. *Biomechanics in Medicine, Sport and Biology*. Springer, 2022.

[31] Larry L Hench and June Wilson. *An introduction to bioceramics*. Vol. 1. World scientific, 1993.

[32] H Hermawan, D Dubé, and D Mantovani. "Developments in metallic biodegradable stents". In: *Acta biomaterialia* 6.5 (2010), pp. 1693–1697.

[33] American Association of Hip and Knee Surgeons. *What are hip and knee replacement implants made of?* URL: https://hipknee.aahks.org/what-are-hip-and-knee-replacement-implants-made-of/.

[34] "HipXNet: Deep Learning Approaches to Detect Aseptic Loos-Ening of Hip Implants Using X-Ray Images". In: *IEEE Access* 10 (2022), pp. 53359–53373. DOI: 10.1109/access.2022.3173424.

[35] Lei Huang et al. "Normalization techniques in training dnns: Methodology, analysis and application". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[36] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[37] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models". In: *International journal of computer vision* 1.4 (1988), pp. 321–331.

[38] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* 30 (2017).

[39] Mohammed Khouy et al. "Medical Image Segmentation Using Automatic Optimized U-Net Architecture Based on Genetic Algorithm". In: *Journal of Personalized Medicine* 13.9 (Aug. 2023), p. 1298. ISSN: 2075-4426. DOI: 10.3390/jpm13091298. URL: http://dx.doi.org/10.3390/jpm13091298.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[41] M VENKATA PAVAN Kumar and I Bhanulatha. "Design and structural analysis of knee implants using different materials". In: *INTERNATIONAL JOURNAL* 5.7 (2020).

[42] Paras Lakhani and Baskaran Sundaram. "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks". In: *Radiology* 284.2 (2017), pp. 574–582.

[43] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[44] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42 (2017), pp. 60–88.

[45] Don P Mitchell. “Generating antialiased images at low sampling densities”. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 1987, pp. 65–72.

[46] Tom M Mitchell. *Machine learning*. 1997.

[47] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[48] Ziad Obermeyer and Ezekiel J Emanuel. “Predicting the future—big data, machine learning, and clinical medicine”. In: *The New England journal of medicine* 375.13 (2016), p. 1216.

[49] Nobuyuki Otsu et al. “A threshold selection method from gray-level histograms”. In: *Automatica* 11.285-296 (1975), pp. 23–27.

[50] SGOPAL Patro and Kishore Kumar Sahu. “Normalization: A preprocessing stage”. In: *arXiv preprint arXiv:1503.06462* (2015).

[51] Dzung L Pham, Chenyang Xu, and Jerry L Prince. “Current methods in medical image segmentation”. In: *Annual review of biomedical engineering* 2.1 (2000), pp. 315–337.

[52] R M Pilliar, J M Lee, and C Maniatopoulos. “Observations on the effect of movement on bone ingrowth into porous-surfaced implants”. en. In: *Clin. Orthop. Relat. Res.* 208.208 (July 1986), pp. 108–113.

[53] James Quinn et al. “Titanium for orthopedic applications: an overview of surface modification to improve biocompatibility and prevent bacterial biofilm formation”. In: *IScience* 23.11 (2020).

[54] Pranav Rajpurkar et al. “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning”. In: *arXiv preprint arXiv:1711.05225* (2017).

[55] Buddy D Ratner et al. *Biomaterials science: an introduction to materials in medicine*. Elsevier, 2004.

[56] Weibin Rong et al. “An improved CANNY edge detection algorithm”. In: *2014 IEEE international conference on mechatronics and automation*. IEEE. 2014, pp. 577–582.

[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[58] Abhijit Guha Roy et al. “Bayesian QuickNAT: Model uncertainty in deep whole-brain segmentation for structure-wise quality control”. In: *NeuroImage* 195 (2019), pp. 11–22.

[59] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[60] R Shashidhar et al. “Advancing Medical Imaging: A Focus on Efficient Net for Brain Tumor Classification”. In: *2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*. IEEE. 2024, pp. 1–5.

[61] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48.

[62] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[63] Carlo Tomasi and Roberto Manduchi. “Bilateral filtering for gray and color images”. In: *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, pp. 839–846.

[64] M Tubiana. “Wilhelm Conrad Röntgen et la découverte des rayons X”. In: *Bulletin de l’Academie nationale de medecine* 180 (1996), pp. 97–108.

[65] Guotai Wang et al. “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks”. In: *Neurocomputing* 338 (2019), pp. 34–45.

[66] Andrew Warburton et al. “Biomaterials in spinal implants: a review”. In: *Neurospine* 17.1 (2020), p. 101.

[67] Gautam P Yagnik et al. “A biomechanical comparison of new techniques for distal clavicular fracture repair versus locked plating”. In: *Journal of Shoulder and Elbow Surgery* 28.5 (2019), pp. 982–988.

[68] Jianpeng Zhang et al. “Covid-19 screening on chest x-ray images using deep learning based anomaly detection”. In: *arXiv preprint arXiv:2003.12338* 27.10.48550 (2020).

[69] Zongwei Zhou et al. “Unet++: A nested u-net architecture for medical image segmentation”. In: *Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, proceedings 4*. Springer. 2018, pp. 3–11.

[70] Djemel Ziou and Salvatore Tabbone. “Edge Detection Techniques-An Overview”. In: *P &#1085 / Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications* 8.4 (1998). Article dans revue scientifique avec comité de lecture., pp. 537–559. URL: `https://inria.hal.science/inria-00098446`.