Degree Project in Medical Engineering

Second cycle, 30 credits

Clinical Assessment of Deep Learning-Based Uncertainty Maps in Lung Cancer Segmentation

FEDERICA CARMEN MARUCCIO

Turin, Italy, 2023

Clinical Assessment of Deep Learning-Based Uncertainty Maps in Lung Cancer Segmentation

FEDERICA CARMEN MARUCCIO

Master's Programme, Medical Engineering, 120 credits
Double Degree Programme, Polytechnic University of Turin

Date: June 21, 2023

Supervisors: Pavlos Papaconstandopoulos, Filippo Molinari, Massimo Salvi
Examiner: Sebastiaan Meijer
School: Engineering Sciences in Chemistry, Biotechnology and Health
Host company: Koninklijke Philips N.V.

Swedish title: Klinisk Bedömning av Djup Inlärningsbaserade Osäkerhetskartor vid Segmentering av Lungcancer

Alla piccola e dolce Manuela,
il cui sorriso ha illuminato la mia vita.

Abstract

Prior to radiation therapy planning, tumours and organs at risk need to be delineated. In recent years, deep learning models have opened the possibility of automating the contouring process, speeding up the procedures and helping clinicians. However, deep learning models, trained using ground truth labels from different clinicians, inevitably incorporate the human-based inter-observer variability as well as other machine-based uncertainties and biases. Consequently, this affects the accuracy of segmentation, representing the primary source of error in contouring tasks. Therefore, clinicians still need to check and manually correct the segmentation and still do not have a measure on the reliability. To tackle these issues, researchers have shifted their focus to the topic of probabilistic neural networks and uncertainties in deep learning models. Hence, the main research question of the project is whether a 3D U-Net neural network trained on CT lung cancer images can enhance clinical contouring practice by implementing a probabilistic auto-contouring system. The Monte Carlo dropout technique was employed to generate probabilistic and uncertainty maps. The model calibration was assessed using reliability diagrams, and subsequently, a clinical experiment with a radiation oncologist was conducted. To assess the clinical validity of the uncertainty maps two novel metrics were identified, namely mean uncertainty (MU) and relative uncertainty volume (RUV). The results of this study demonstrated that probability and uncertainty mapping effectively identify cases of under or over-contouring. Although the reliability analysis indicated that the model tends to be overconfident, the outcomes from the clinical experiment showed a strong correlation between the model results and the clinician's opinion. The two metrics exhibited promising potential as indicators for clinicians to determine whether correction of the predictions is necessary. Hence, probabilistic models revealed to be valuable in clinical practice, supporting clinicians in their contouring and potentially reducing clinical errors.

Keywords

3D U-Net, Contouring, Clinical validation, Deep learning, Lung cancer, Monte Carlo dropout, Probability map, Reliability diagram, Segmentation, Uncertainty map

Sammanfattning

Innan planering av strålbehandling måste tumörer och riskorgan avgränsas. Under de senaste åren har djupinlärningsmodeller öppnat upp för möjligheten att automatisera kontureringsprocessen, vilket påskyndar åtgärderna och hjälper läkarna. Djupinlärningsmodeller som tränas med hjälp av grund sanning från olika läkarna, innehåller dock oundvikligen den människobaserade variabiliteten mellan observatörer samt andra maskinbaserade osäkerheter och fördomar. Detta påverkar följaktligen segmenteringens noggrannhet, som utgör den främsta felkällan i kontureringsuppgifter. Därför måste läkarna fortfarande kontrollera och korrigera segmenteringen manuellt. För att ta itu med dessa problem har forskarna flyttat sitt fokus till ämnet probabilistiska neurala nätverk och osäkerheter i modeller för djupinlärning. Projektets viktigaste forskningsfråga är därför om ett probabilistiskt system för automatisk konturering kan förbättra klinisk konturering. För att besvara denna fråga utvecklades och tränades ett 3D U-nät neuralt nätverk med hjälp av CT lungcancer bilder. Monte Carlo Dropout-tekniken användes sedan för att generera probabilistiska och osäkerhetskartor. Modellkalibreringen bedömdes med hjälp av tillförlitlighetsdiagram och därefter genomfördes ett kliniskt experiment med en strålningsonkolog. För att bedöma osäkerhetskartornas kliniska giltighet identifierades två mått, nämligen medelosäkerhet (MU) och relativ osäkerhetsvolym (RUV). Resultaten av denna studie visade att sannolikhets- och osäkerhetskartläggning effektivt identifierar fall av under- eller överkonturering. Även om tillförlitlighetsanalysen visade att modellen är överdrivet självsäker för höga osäkerhetsvärden, visade resultaten från det kliniska experimentet en stark korrelation mellan modellens resultat och läkarens åsikt. De två mätvärdena uppvisade en lovande potential som indikatorer för läkarna avgöra om det är nödvändigt att korrigera prognos. Sannolikhetsmodeller visade sig därför vara värdefulla i klinisk praxis genom att stödja kliniker i deras konturering och potentiellt minska kliniska fel.

Nyckelord

3D U-Nät, Konturering, Klinisk validering, Djupinlärning, Lungcancer, Monte Carlo dropout, Sannolikhetskartor, Tillförlitlighetsdiagram, Segmentering, Osäkerhetskartor

Acknowledgments

First, I would like to thank Philips Research for giving me the privilege of being part of their innovative company and for allowing me to collaborate with incredible colleagues throughout this enriching experience. The knowledge and skills I gained from working alongside them have been invaluable. They taught me the true meaning of working with passion and dedication, as well as the importance of mutual support and collaboration. In particular, I am immensely grateful to my supervisor at Philips, Pavlos Papaconstadopoulos, for his guidance, support and encouragement throughout this research. I attribute the achievement of the project goals as well as my personal and professional growth to his exceptional expertise and unwavering dedication. I extend my sincerest thanks to Max-Heinrich Laves, Roger Fonolla Navarro, Wietse Eppinga, Edwin Heijman and all the other colleagues in my department, who were always ready to help me without hesitation in the completion of this project.

Furthermore, I express my appreciation to the professors from Politecnico di Torino and Kungliga Tekniska högskolan in Stockholm, specifically Filippo Molinari, Massimo Salvi, Maksims Kornevs, Jayanth Raghothama and Sebastiaan Meijer, for their helpful contributions throughout the project and thesis composition. I would also like to acknowledge the guidance provided by professor Valentina Agostini, who mentored me as a double degree student.

This project stands as the crowing achievement of a long journey of study and growth which has shaped me on both personal and professional levels. I feel immensely lucky for the opportunity given by EDISU to live, study and be financially autonomous throughout my whole educational path. I deeply cherish the hope that the right to pursue education will be accessible to all individuals without any denial or discrimination.

As a final remark, I would like to acknowledge my family and friends for their immense support and continuous belief in me during all the past years.

Above all, I am deeply grateful to my dad Graziano, my mom Barbara and my brother Alessandro, for their constant encouragement and unconditional love demonstrated daily even from miles away. A special mention is deserved to my little sister Greta, who walked alongside me for a significant portion of this journey and represented a constant source of strength and motivation. I extend my acknowledge to my grandmother Adriana, who has been always present in my life, proving that love can overcome all barriers.

An important acknowledge belongs to my lifelong friends, Chiara, Elena, Federico and all the others who have always been there for me when I returned

back home, holding an unaltered friendship even after years of physical distance. In particular, I would like to thank a special person, Maria Grazia, who has taught me the true essence of genuine friendship.

This journey, which started in Turin, allowed me to meet incredible people who built a second family around me throughout my time in that beautiful city and even after. Above all, Marianna, Manuel, Mattia and Gianmarco, whose presence has been, and still is, truly precious for me. A particular remark goes to Simone, whose passion and willingness to learn has always motivated me and given me inspiration.

Moreover, I consider myself incredibly privileged to have had the incredible opportunity of living in Stockholm for a year, where I cultivated some of the most authentic friendships and encountered the most intense emotions of my life. I am particularly grateful to Sofia, Simone, Gianluca, Lucia, Ludovica and Ilaria who taught me that deep bonds can be forged even within a short span of a few months. This experience also allowed me to meet one of the most sincere friend I have ever had, Dario, who has become a significant anchor in my life.

My third and last big adventure has been moving to Eindhoven, where I concluded my journey as a student. In only 9 months I had the chance to get to know fantastic people including Marco, Yarib, Alessandra, and many others who supported me throughout my stay. A major acknowledgement is reserved for my partner in crime, Martina, who has been my stable reference and continuous wellspring of joy since my earliest days in The Netherlands.

Last but not the least, I express my warmest thanks to the person who, above all, has been undeniably essential and unwaveringly supportive throughout the past six years, consistently celebrating my accomplishments and providing steadfast help during challenging times. Gianmarco, who holds the most significant place in my heart, has undoubtedly played a crucial role in making this achievement possible.

Stockholm, June 2023
Federica Carmen Maruccio

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

3D	Three-Dimensional
AI	Artificial Intelligence
AUC	Area Under the ROC Curve
BN	Batch Normalization
CNN	Convolutional Neural Network
Conv	Convolution
ConvTransp	Transposed Convolution
CT	Computed Tomography
DICOM	Digital Imaging and COmmunications in Medicine
DL	Deep Learning
EBUS	EndoBronchial UltraSound
ECE	Expected Calibration Error
ETL	Extract Transform Load
FCN	Fully Convolutional Network
GTV	Gross Tumor Volume
HDF5	Hierarchical Data Format version 5
HU	Hounsfield unit
MaxPool	Max Pooling
MC	Monte Carlo
ML	Machine Learning
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MU	Mean Uncertainty
NN	Neural Network
NSCLC	Non-Small Cell Lung Cancer

OAR	Organs At Risk
p	dropout rate
PET	Positron Emission Tomography
ReLu	Rectifier Linear Unit
RO	Radiation Oncologist
ROC	Receiver Operating Characteristic
RT	Radiation Therapy
RUV	Relative Uncertainty Volume
SCLC	Small Cell Lung Cancer
SGD	Stochastic Gradient Descent
TNM	Tumor-Node-Mestastasis
UCE	Expected Uncertainty Calibration Error
VAT	Video-Assisted Thoracoscopy
WHO	World Health Organization

Chapter 1

Introduction

Deep neural networks have demonstrated impressive results in various computer vision tasks. However, in real-world applications such as autonomous driving, medical diagnoses, and nuclear power plant monitoring, concerns about safety have arisen due to the potential severe consequences of failures [1, 2]. These networks are often referred to as 'black box' models, which may fail silently without indicating that the prediction is incorrect. This is especially concerning for safety-critical applications in medicine, such as neurosurgical interventions and radiotherapy planning [3, 4].

In the specific field of medical image segmentation, the primary challenge leading to incorrect segmentations is the high inter-observer variability. In Machine Learning (ML) models, a common practice is to generate ground truth labels by merging multiple observations from different clinicians. However, different doctors may delineate very different areas due to factors such as image quality, personal bias, level of expertise, and knowledge [5]. As a result, deep learning models trained with this data include human-based inter-observer variability, as well as other machine-based uncertainties and biases due to the model architecture and parameters. This affects the accuracy of the segmentation and is the primary source of error in contouring tasks. This raises the critical question: how can we trust the predictions of these models?

To address these issues, researchers have shifted their focus to the topic of probabilistic neural networks and uncertainties in artificial intelligence models. Thus, the main research question of the project is the following:
Can a probabilistic auto-contouring system improve clinical contouring practice?

To increase trust and interpretability in deep learning algorithms [6] for medical imaging analysis, we need tools such as uncertainty estimates that

can improve the robustness of automated segmentation systems. By providing clinicians with an uncertainty map, they can better understand where and why the model failed and manually correct the segmentation [3].

Gal and Ghahramani [7] demonstrated that model uncertainty can be obtained from dropout neural networks. In the field of deep learning, dropout refers to the training procedure whereby a subset of nodes and their connections in a neural network are randomly suppressed to avoid overfitting and improve generalisation. The authors suggested using dropout during the testing phase as well in order to produce different segmentations that can be averaged together. By utilising the so-called Monte Carlo dropout technique [8], probability and uncertainty maps can be generated on images during prediction. These maps can be valuable in the field of radiation therapy by aiding clinicians in their contouring task. Probability maps can indicate which areas on the image have a high probability of being a tumor and should be included in the target area. Uncertainty maps will highlight regions with high uncertainties, warning clinicians to exercise caution when contouring those areas.

The objective of this project is to make a contribution to the field of oncological image analysis by proposing the application of Monte Carlo dropout technique in clinical practice. The project will demonstrate the benefits of utilising probability and uncertainty maps for the clinical case of lung cancer patients.

1.1 Purpose

The primary objective of the project is to address the inherent reliability issue in deep learning models for segmentation. Currently, artificial intelligence tools cannot be entirely relied upon for high-risk tasks, and thus require a clinician's evaluation of the model's results. The main goal of this project is to develop a methodology that utilises uncertainty and probability maps as supplementary tools for the clinician to assess the segmentation quality. By doing so, high-uncertainty segmentations can be reviewed by the clinician, while low-uncertainty delineations can be automatically approved, resulting in significant time savings. Furthermore, the use of uncertainty maps can help address undercontouring and overcontouring issues by identifying areas of uncertainty and alerting the clinician. This can potentially help prevent undertreatment or toxicity issues.

1.2 Goals

The final goals of the project from a general point of view can be summarised in four main points:

- Build a 3D U-Net able to produce segmentations of the tumor;
- Implement Monte Carlo dropout technique to develop probability and uncertainty maps;
- Check reliability of the probabilistic model using reliability diagrams;
- Clinically validate the model by the expert.

1.3 Research Methodology

The methodology adopted in this project follows the main steps of developing a standard neural network with a few extra steps to make the model capable of producing probability and uncertainty maps. Each step is listed in the following:

- Understand the nature of the problem and the state of the art in medical image segmentation and uncertainty maps by performing a detailed literature review;
- Set-up the infrastructure needed for training and running neural networks on a Linux cluster.
- Model a 3D U-Net neural network on the Linux cluster capable of generating auto-contours given medical images;
- Train, validate and test the performance of the network;
- Expand the model in order to generate multiple segmentations using Monte Carlo dropout technique and aggregate solutions to generate probabilistic and entropy maps;
- Extract and evaluate model calibration curves per structure by expected frequencies to probabilities;
- Clinical validation of the model through a blind experiment designed to compare uncertainty scores provided by the model with the clinician's opinions.

This approach ensures to start with a simple, known neural network and then enrich the model step by step with the goal of obtaining a probabilistic auto-contouring system for medical purposes.

Figure 1.1 shows the project pipeline and summarises the steps described previously.

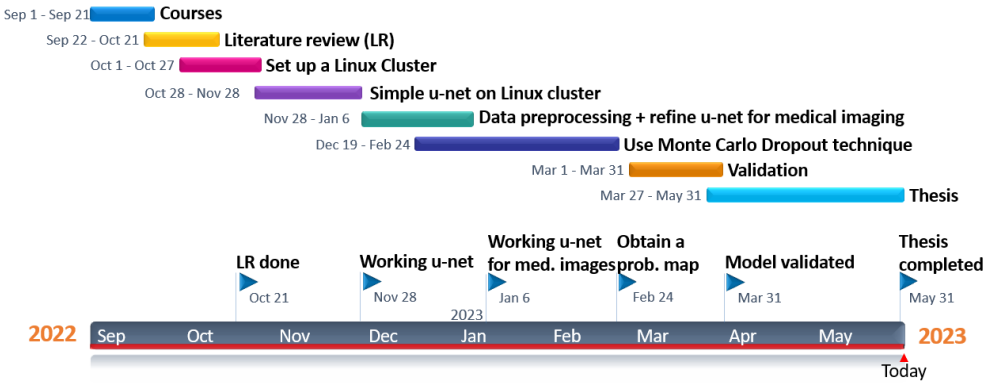


Figure 1.1: Project timeline.

1.4 Limitations

The primary objective of this project is to utilise the Monte Carlo dropout technique to develop probability and uncertainty maps. It is crucial to note that the dataset that has been used to train the model was quite small in terms of number of patients and with a high variability of the tumor. The segmentation performance achieved with this project reflects the averaged results of other works that used the same dataset [9]. However, the key focus of the project is on the second stage of the pipeline, specifically during testing where new innovations were implemented. As a result, the outcomes from the first stage have a minimal impact on the primary accomplishments of the project.

1.5 Structure of the Thesis

The thesis is divided into the following chapters:

- Introduction 1, which presents an overview of the research goals to the reader;

- Background 2, which provides fundamental information related to previous research and the theory behind medical image segmentation and deep learning tools;
- Methodology 3, which explains the choices and steps taken during the project to reach the goals;
- Results 4, which shows the outcomes obtained, in particular regarding probability and uncertainty maps and clinical validation;
- Discussion 5, which analyses and interprets the results;
- Conclusion 6, which highlights the research objectives achieved and describes any limitations and future developments.

Chapter 2

Background

The background chapter introduces two distinct fields, medicine and computer science. The chapter covers an extensive range of topics, including lung cancer, radiotherapy, and medical image segmentation, in addition to the fundamental concepts of deep learning. Furthermore, the chapter provides a review of the implementation of probability and uncertainty mapping techniques, along with strategies for validating the reliability of these tools.

2.1 Lung Cancer

This section covers various aspects of lung cancer, including its epidemiology and associated risk factors, staging and classification, as well as diagnosis and treatment methods.

2.1.1 Lung Cancer Epidemiology and Risk Factors

Although the incidence and mortality of lung cancer have been consistently decreasing in the last decade, it still maintains its position as the leading cause of cancer-related deaths [10]. Lung cancer is responsible for nearly 25% of all cancer-related deaths, with 82% of these deaths directly linked to cigarette smoking [11].

It ranks second in terms of incidence, with prostate cancer being the only more common cancer in men and breast cancer in women. While lung cancer is more prevalent in men, the rate of decline in lung cancer incidence is slower for women when compared to men.

Patients under the age of 40 have a relatively low incidence of lung cancer, which gradually increases and reaches its peak between the ages of 65 and 84

[12].

There are various well-established risk factors for developing lung cancer, [10]. The most important ones are listed below in order of relevance [10, 13]:

- **Cigarette smoking**, representing the number one risk factor;
- Exposure to **second-hand smoke**;
- **Environmental hazards**, such as asbestos and radon, due to occupational exposure;
- **Air pollution**, including emissions rich in polycyclic aromatic hydrocarbon compounds;
- Personal or **family history** of lung cancer;
- **Dietary habits** and supplements.

2.1.2 Lung Cancer Classification and Staging

Lung cancer can be classified into two main forms: **Non-Small Cell Lung Cancer (NSCLC)** and **Small Cell Lung Cancer (SCLC)**, with NSCLC accounting for 85% of patients. The primary distinguishing factor between the two principal types of lung cancer is the cellular morphology observed through a microscope. SCLC cells exhibit a flatter appearance and smaller size compared to cancer cells present in NSCLC. SCLC usually progresses more quickly than NSCLC and has a tendency to spread to the lymph nodes. Approximately 75% of the patients diagnosed with SCLC already have advanced-stage cancer. NSCLC typically spread at a slower pace and generates fewer symptoms [14].

The World Health Organization (WHO) classifies NSCLC into three types: adenocarcinoma, squamous cell carcinoma, and large cell. Adenocarcinoma represents the most common type of NSCLC, while squamous cell carcinoma represents 25% to 30% of lung cancers. Large cell cancers account for approximately 5% to 10% of all lung cancers. The WHO also recognises early stages of lung cancer as adenocarcinoma in-situ, minimally invasive adenocarcinoma, or invasive adenocarcinoma based on the extent of invasiveness. Immunohistochemical markers are typically present and used for the diagnosis and characterization of these types of lung cancer [12]. After the initial diagnosis of NSCLC, accurate staging using the **Tumor-Node-Mestastasis (TNM)** classification system is essential for selecting the

appropriate therapy after diagnosis, predicting prognosis, and evaluating the response to treatment [13]. The TNM system involves evaluating the dimension of the primary tumour (T), regional lymph node(s) involvement (N), and distant metastases (M) [15]. After determining T, N, and M, they are merged to assign an overall stage of 0, I, II, III, or IV. These stages can be also further classified using letters, like IIIA and IIIB.

2.1.3 Lung Cancer Diagnosis

The typical procedure for the diagnosis of lung cancer starts with the patient recognizing suspicious symptoms such as cough, hemoptysis, chest pain, and dyspnea.

The first step in diagnosing lung cancer is typically through the use of **imaging tools** such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET) scans. The first imaging test is performed with posteroanterior and lateral chest radiography. Then, all suspected patients undergo a chest CT scan to define the size, shape and location of the lesion. PET is used to provide information about the metabolic nature of the lung tumour by injecting into the body 18-fluorodeoxyglucose (FDG) to recognise cancer cells. Studies confirmed that integrated PET/CT scanners appear to produce more detailed images than CT or PET alone [16]. Abdominal CT, bone scanning, and brain MRI are usually recommended in patients with small cell carcinoma because of the high likelihood of metastatic disease.

After locating the lesion using imaging methods, the next phase is to determine the appropriate technique to get a **biopsy sample** for histologic confirmation [17]. Biopsies are the most commonly used tool to obtain tissue for diagnosing lung cancer, which may include needle biopsy, EndoBronchial UltraSound (EBUS), mediastinoscopy, Video-Assisted Thoracoscopy (VAT), and wedge resection [18].

2.1.4 Lung Cancer Treatment

The type of cancer, the stage at diagnosis, and the patient's functional assessment are factors that determine the treatment approach. NSCLC patients in stages I to IIIA are typically treated with **surgery**, which could be lobectomy (surgical resection of a lobe) or sub-lobar resection. Recent evidence indicates that preoperative chemotherapy can increase survival rates. For those who undergo complete resection without preoperative chemotherapy, standard

treatment involves adjuvant chemotherapy.

Unresectable non-small cell carcinoma patients may be treated with Radiation Therapy (RT) and chemotherapy. High-energy X-rays with strong killing or growth-inhibiting properties are employed in lung cancer **RT** to destroy cancer cells. External radiation, which involves administering radiation from outside the body, is more frequently utilized than internal or implant radiation, which involves using radioactive materials placed directly inside the lung cancer tumour [19]. On the other hand, **chemotherapy** employs powerful drugs that travel through the body's bloodstream and target cancer cells [20].

In salvage therapy after surgery, RT, or chemotherapy or for palliation in advanced NSCLC, percutaneous **thermal ablation** procedures such as cryoablation, microwave, and radiofrequency ablation have been considered effective treatment options [12].

2.1.4.1 Radiotherapy

As previously stated, external beam RT is the primary treatment modality for inoperable lung cancer patients often combined with chemotherapy [21]. The workflow of RT typically involves several stages, which are described below and illustrated in Figure 2.1:

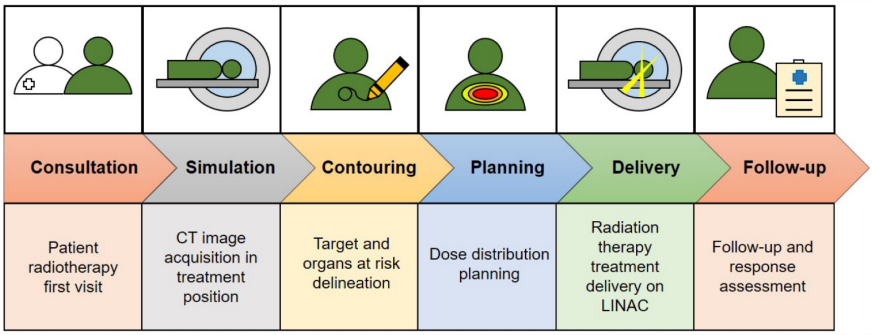


Figure 2.1: Radiotherapy workflow [22] (CC BY 4.0).

- **Consultation**: The patient is referred to a Radiation Oncologist (RO) by their primary care physician. The RO will review the patient's medical history and imaging studies, and determine if radiotherapy is an appropriate treatment option.
- **Simulation**: If radiotherapy is considered appropriate, the patient will undergo a simulation procedure to help the RO determine the best

treatment plan. It involves the use of imaging techniques, such as CT or MRI scans, to precisely localize the tumour and surrounding healthy tissues, as well as to determine the appropriate techniques for treatment delivery.

- **Contouring**: It involves the delineation of the tumour and normal tissues on the imaging data, in order to accurately define the Gross Tumor Volume (GTV) (visible extent of the tumour in the medical image) and Organs At Risk (OAR) for radiation planning.

- **Planning**: Specialized software are used to develop a radiation treatment plan that takes into account the individual patient's anatomy, tumour characteristics, and treatment goals while minimizing the dose to surrounding healthy tissues.

- **Treatment delivery**: Once the plan has been developed, the patient will begin receiving radiotherapy treatments. These may be delivered using external beam RT, in which a machine called a linear accelerator delivers the radiation beams from outside the body, or internal RT, in which radioactive sources are placed inside the body in or near the tumour. The number and duration of treatments will depend on the specific treatment plan, but most patients receive treatment on a daily basis for several weeks.

- **Follow-up**: After the RT treatment is completed, the patient will typically have regular follow-up visits with their RO to monitor their progress and evaluate any potential side effects or complications. These visits may involve imaging studies, blood tests, and physical exams.

Overall, the RT workflow is a carefully orchestrated process that involves multiple stages of planning and delivery, with the ultimate goal of effectively treating the patient's cancer while minimizing the risk of side effects and complications. It is also clear that the contouring step is crucial in order to avoid mistreatment due to over or under-irradiation of the GTV and OAR.

2.2 Medical Image Segmentation

Image segmentation is a crucial step in medical image processing that involves dividing a digital image into multiple segments, each comprising sets of voxels. There are three main categories of image segmentation techniques:

- **manual** segmentation, which is carried out by a RO who annotates the voxels of interest manually;
- **semi-automatic** segmentation, which involves algorithms that aid in the segmentation process or help finalize the contouring,
- **automatic** segmentation, which does not require user input and can be classified into learning and non-learning-based methods [23].

However, the RO is the clinician responsible for accepting the GTV delineation, even if automatic segmentation has been used. This is because reliable GTV auto-segmentation models are yet to be developed, with the current models primarily focusing on accurate segmentation of OARs.

Conventional automatic segmentation methods rely on the surface-level characteristics of the image, including grayscale, texture, and gradient, to segment the desired target. Thresholding Method, Atlas Method, and Region Growing Method are some of the common approaches employed in traditional automatic segmentation techniques [24].

The **Thresholding Method** selects appropriate grayscale thresholding based on the target and background that require segmentation. Subsequently, all pixels in the image are categorized into either the target or background group.

On the other hand, the **Atlas Method** aligns the new input image with a reference image known as an atlas template. The labels in the atlas are then applied to the new input image to accomplish the segmentation task.

The **Region Growing Method** involves the manual delineation of sub-regions, followed by the merging of neighbouring pixels with similar attributes into the predetermined region and segmenting the target area from the background.

As **Deep Learning (DL)** technology advances, models based on DL have demonstrated remarkable potential in auto-segmentation of medical images. DL models independently learn feature representation and utilize the acquired high-dimensional abstraction to segment without the need for manual interaction [24].

2.2.1 Segmentation in Radiotherapy

In the field of RT, precise targeting of the **GTV** and protecting **OAR** from radiation-related complications are crucial for its effectiveness. Three main volumes are to be considered in RT, as illustrated in Figure 2.2. The

GTV represents the position and extent of the primary tumour, the **clinical tumour volume (CTV)** encompasses the GTV and defines the extent of microscopic cancerous spread that can't be seen on imaging; the **planning target volume (PTV)** is added to account for uncertainties in planning or delivery [25]. In Figure 2.3, an example of segmentation of OAR and GTV is illustrated. Accurate segmentation of the GTV and OAR is essential in

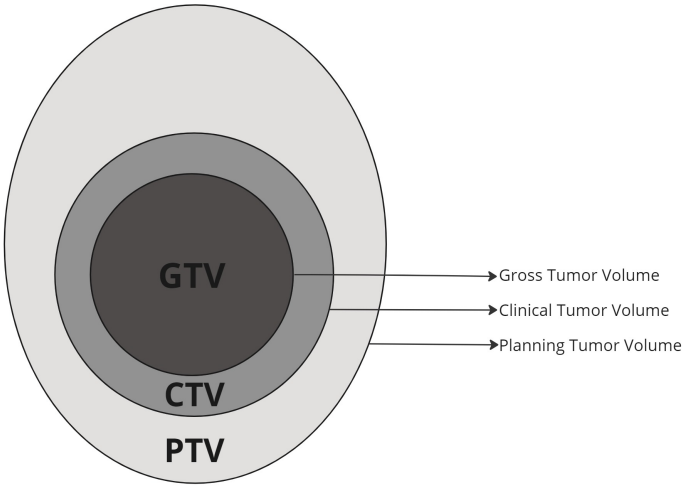


Figure 2.2: Diagram of the main radiotherapy planning volumes.

RT treatment planning to deliver the intended dose to the GTV. However, as previously stated, manually segmenting the GTV and OAR is a tedious and time-consuming task for ROs. This can cause delays in treatment and adversely affect survival rates. Moreover, the quality of manual segmentation is subject to the ROs' expertise. Even when following the same guidelines, inconsistencies in the segmentation may occur among both inter- and intra-observers [24].

Accurate delineation of the GTV and OAR is critical to avoid OAR over-irradiation while still treating effectively the GTV. Even a small error in the segmentation, such as a 1 mm shift, could have a significant impact on radiotherapeutic dose calculations, with an estimated effect of up to 15% [26].

The use of DL in clinical practice for RT has the potential to reduce unnecessary time and relieve relevant staff of their workload, thus avoiding errors caused by fatigue [24]. Moreover, while other machine learning methods could be more interpretable and easy to implement, DL's ability to automatically extract and learn complex features, scalability to large datasets, and integration of multimodal data make it a particularly powerful approach for tumour segmentation in oncology studies.

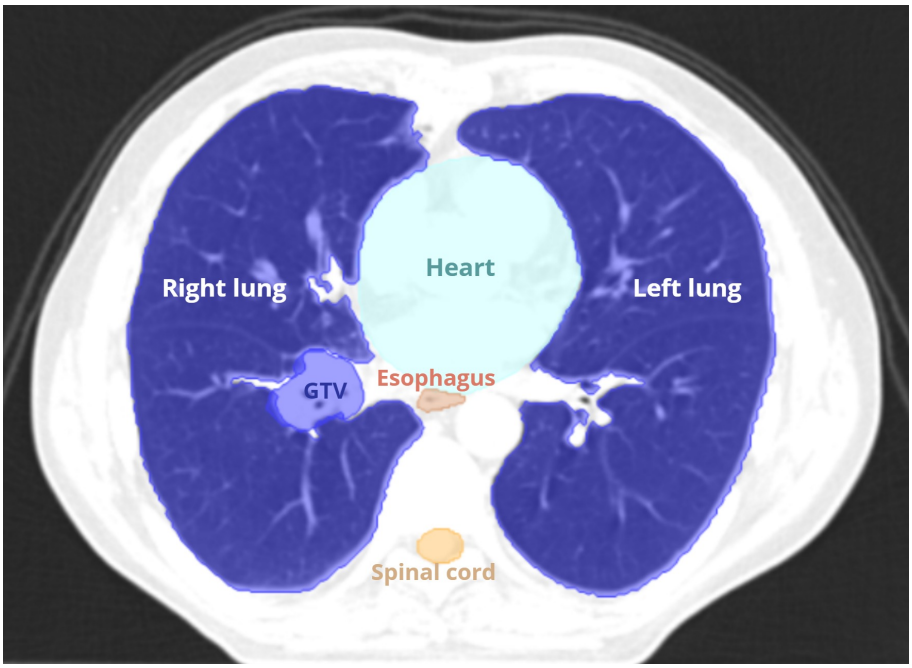


Figure 2.3: Example of organs at risk and GTV delineations for lung cancer.

2.3 Deep Learning in Medical Image Segmentation

This section will first present an overview of the fundamental concepts of DL. Following that, the focus will shift to the field of image segmentation, with a detailed description of the most commonly employed DL model in medical image segmentation.

2.3.1 Machine Learning and Neural Networks

Before delving into the discussion of deep learning, it is essential to establish a foundational understanding of the field by distinguishing between various terms and concepts. To achieve this objective, Artificial Intelligence (AI), ML, and DL concepts are described in the following. The hierarchical relationship between these terms is summarized in the Venn diagram depicted in Figure 2.4 [27].

AI represents the intelligence demonstrated by machines and comprises a wide range of techniques that allow computers to imitate human behaviour and perform complex tasks independently or with minimal human involvement,

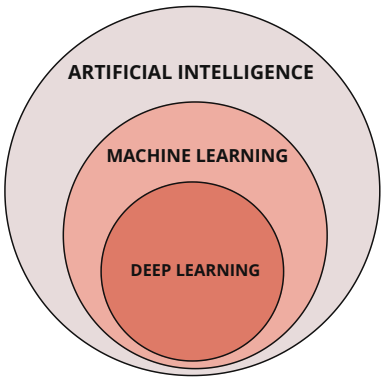


Figure 2.4: Venn diagram illustrating the hierarchical relationship between AI, ML and DL.

sometimes even surpassing human decision-making abilities. **ML** refers to the field devoted to enhancing a computer program's ability to perform a particular set of tasks. This is accomplished by utilizing algorithms that learn from training data related to the specific problem, enabling computers to identify patterns and hidden insights without the need for explicit programming [27]. DL, a subset of ML techniques, relies on artificial neural networks to process information.

The architecture of DL draws inspiration from the information processing principles of the biological neural network found in the human brain. A biological neuron serves as the fundamental building block of the human neural network and comprises three major components, as shown in Figure 2.5:

- a cell body (soma), which houses the nucleus and other cell-supporting structures,
- dendrites, that receive electro-chemical signals from neighbouring neurons
- an axon, that transmits the signal to the following neuron [28].

Dendrites of the two adjacent neurons are connected through a synapse. If the signal received is strong enough to surpass a certain threshold, the neuron is triggered, and the signal is transmitted to the subsequent neuron.

An artificial neuron tries to imitate the behaviour and structure of biological neurons, and neural networks resemble the brain in two key aspects:

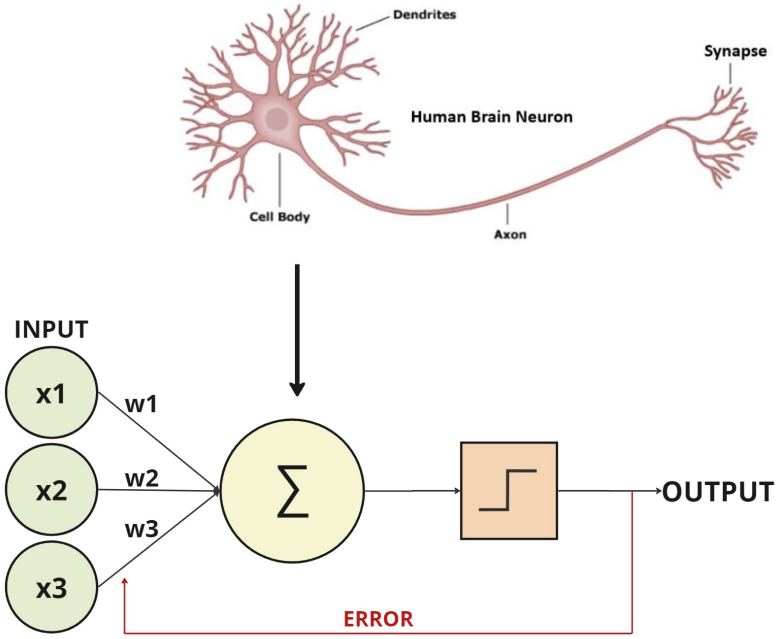


Figure 2.5: Comparison between a biological neuron and an artificial neuron's structure. Adapted from [29].

- They acquire knowledge through a learning process;
- The knowledge is stored using synaptic weights.

A learning algorithm modifies the connection weights between neurons to achieve a specific goal. There are two types of learning:

- **Supervised learning**: the weights are adjusted in order to minimize the error between the output and the given target;
- **Unsupervised learning**: the aim is to cluster dataset elements in homogeneous groups.

The simplest artificial neural network, the perceptron shown in Figure 2.5, classifies inputs into one of two classes using an activation function that receives the weighted sum of inputs. However, since a **perceptron** is a single-layer Neural Network (NN), it cannot perform non-linear classification, which can be overcome by adding hidden layers to create a deep neural network. A NN that contains one or more hidden layers is called deep neural network and it is organised in a deeply nested architecture. This is the reason why it is called

"deep learning" since several layers are required for processing the data and generating output. For sake of clarity, NN layers are composed of a certain amount of neurons and organized as follows (Figure 2.6):

- **Input layer**: consisting of neurons that receive input from the environment;
- **Output layer**: consisting of neurons that produce the final output of the network and provide it to the environment;
- **Hidden layer(s)**: consisting of neurons that do not have direct contact with the environment.

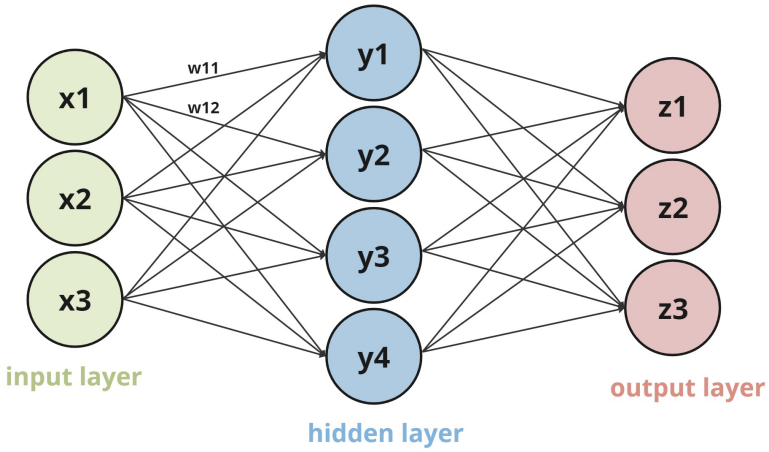


Figure 2.6: Neural Network functioning example.

Neurons of one layer connect fully or partially to those in the closest layers, with different NN types based on the task and the optimal topology.

Learning in an NN can be divided into two phases:

- Input patterns are presented to the input layer, then propagated layer to layer until an output pattern is generated;
- If the generated output pattern differs from the target output, an error is computed and propagated backwards from the output layer to the input layer while modifying the weights accordingly.

2.3.2 Convolutional Neural Networks

Using traditional fully-connected artificial neural networks (ANNs) for image-based real-world problems is challenging due to the large number of network components required, leading to computational complexity and loss of spatial information.

To address these challenges, Convolutional Neural Network (CNN)s are used. CNNs use the convolution operator, which replaces matrix multiplication with a set of convolution kernels. This allows for tractable learning of the kernels and reduces computational complexity [30]. They are widely used for image and video recognition.

The main building blocks of a CNN are convolutional layers, pooling layers, and fully connected layers [30], as illustrated in Figure 2.7. **Convolutional layers** apply a series of filters to an input image, which extract various features from the image. The stride and padding are two essential parameters in convolution. The stride determines how the filter is shifted along the input matrix, while the padding determines whether additional zero-padding is added to the input feature map to ensure that the window can always be centered on the input matrix value. The resulting feature maps are then fed into **pooling layers**, which downsample the feature maps to reduce their size and make the network more efficient. After several convolutional and pooling layers, the output is flattened and fed into **fully connected layers**, also called dense layers, which perform a final classification of the input. These fully connected layers learn to combine the extracted features from earlier layers to make a final prediction.

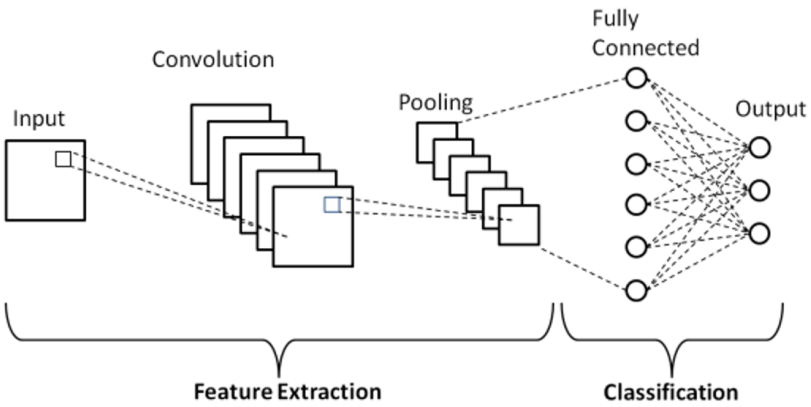


Figure 2.7: Schematic diagram of a basic CNN architecture [31] (CC BY-NC 3.0).

Convolution is a linear operation, hence applying non-linearity is quite necessary to increase the expressive power of the model. Therefore, convolutional layers also incorporate non-linear activation functions such as **Rectifier Linear Unit (ReLu)** to introduce non-linearity into the network.

Batch normalization layers are also used in a CNN architecture to normalize the output of the previous layers by subtracting the mean and dividing by the standard deviation of the activations. This helps to prevent the internal covariate shift problem and improves the stability and speed of training [32].

Moreover, adding a regularization term is important to reduce the risk of overfitting. **Dropout** is a regularization technique that trains only a random subset of neurons at each iteration, so a portion of neurons are randomly dropped and do not contribute to training [33] (Figure 2.8).

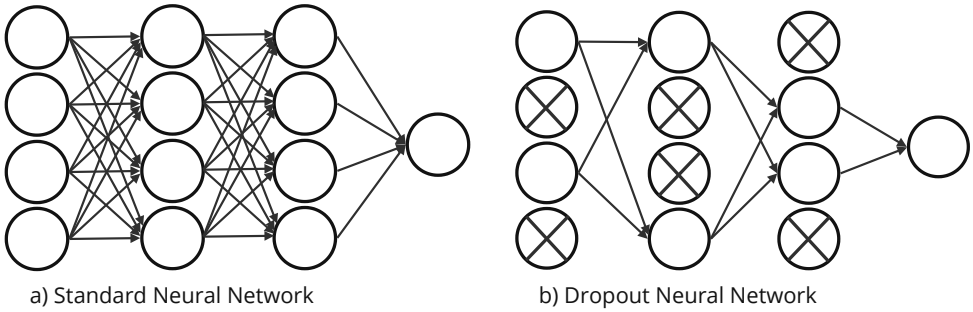


Figure 2.8: Dropout neural network. On the left, no dropout was applied, on the right, dropout was applied and crossed neurons were dropped.

By arranging these fundamental layers in various sequences, several distinct architectures of neural networks can be built.

2.3.2.1 Preprocessing, Training and Performance Assessment

To train a NN, after defining the architecture, it is necessary to prepare the data, split the data set into training, validation, and testing sets, choose an optimization algorithm and a loss function, tune the hyperparameters, and evaluate the performance.

First, it is important to preprocess the data. Medical images are typically stored in **Digital Imaging and COmmunications in Medicine (DICOM) files**, which is the international standard for storing, transmitting, and exchanging medical images and additional information [34]. DICOM files are used in various medical imaging modalities such as X-ray, CT, MRI,

ultrasound, and more. The format allows for the storage of not only the image data but also additional metadata, such as patient information, acquisition parameters, and annotations.

All the DICOM files from the different patients can be combined into a single repository, which could be a Hierarchical Data Format version 5 (HDF5) file [35]. HDF5 is a flexible and efficient file format designed for storing and organizing large and complex datasets. It has a hierarchical structure and it is organized into groups, datasets and attributes, as shown in Figure 2.9.

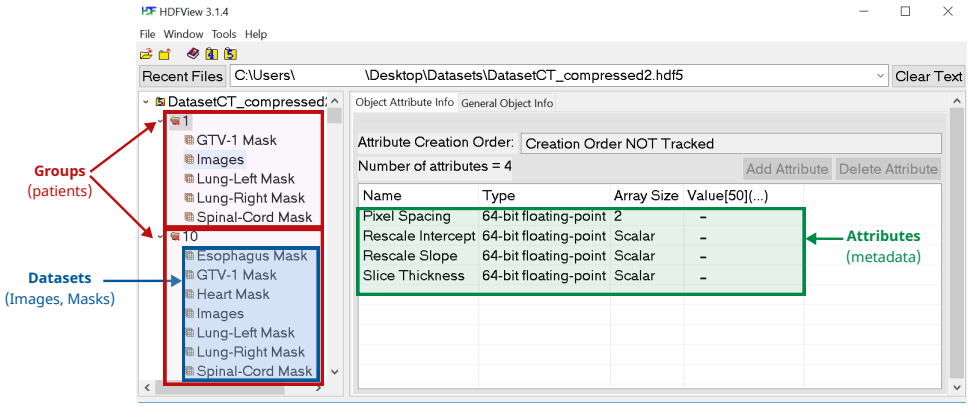


Figure 2.9: Snapshot of the HDF5 View, a visual tool to visualize and edit HDF5 files.

After organizing the file, the dataset needs to be cleaned up and preprocessed. Once the dataset it is ready and split, the model can be trained using the training set, a loss function and an optimization algorithm.

The **loss function** measures the difference between the predicted output of the neural network and the true output, while the **optimization algorithm** is the method for minimizing the loss function during the training process of a neural network. Common optimization algorithms include Stochastic Gradient Descent (SGD), Adam, and Adagrad.

Regarding the hyperparameters, the most important to consider when training a neural network are listed in the following:

- **Learning rate**: determines how quickly the neural network adjusts its weights during training. A high learning rate can cause the weights to oscillate, while a low learning rate can result in slow convergence. It usually ranges between 10^{-2} and 10^{-6}.

- **Batch size**: sets the number of samples used in each iteration of training. Larger batch size can result in faster convergence but can also lead to

overfitting.

- **Number of epochs**: defines the number of times the entire training dataset is passed through the neural network during training. A higher number of epochs can result in better performance, but can also lead to overfitting.

After training the model, each NN segmentation model's performance needs to be assessed. There are several evaluation metrics that can be used, some of which are listed below:

- **Precision**, also called true predictive assessment: the ratio between the number of true positives (TP) to the sum of all the cases reported by the model as positives (True Positive + False Positive).

$$\text{Precision(PR)} = \frac{TP}{TP + FP} \tag{2.1}$$

- **Recall**, also called sensitivity or true positive rate: ratio between the number of true positives to the sum of all the real positive cases (True Positive + False Negative).

$$\text{Recall(RE)} = \frac{TP}{TP + FN} \tag{2.2}$$

- **False Positive Rate**, also specificity: the ratio between the number of false positives to the sum of all the real negative cases (True Negative + False Positive).

$$\text{FPR} = \frac{FP}{TN + FP} \tag{2.3}$$

- **F1 Score**: metric that combines precision and recall in the form of a harmonic mean to measure the overall performance of a segmentation model.

$$\text{F1 score} = \frac{2 * PR * RE}{PR + RE} \tag{2.4}$$

- **Intersection over Union** (IoU), also called Jaccard index: area of intersection between the predicted mask and the ground truth mask divided by the the area of union of the predicted mask and the ground truth mask. The score range is from 0 to 1, where 1 indicates perfect overlap.

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{2.5}$$

- **Dice Score**: one of the most common metrics for segmentation. It measures the similarity between the predicted and ground truth masks. The coefficient ranges from 0 to 1, with 1 indicating perfect similarity.

$$\text{Dice score} = \frac{2|A \cap B|}{|A| + |B|} = \frac{2 * TP}{(TP + FP) + (TP + FN)} \tag{2.6}$$

- **Pixel Accuracy**: measures the percentage of correctly classified pixels in the segmentation mask.

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

Moreover, the performances of a classification model can be visually illustrated using a **Receiver Operating Characteristic (ROC) curve**, as shown in Figure 2.10. This curve plots the true positive rate as a function of the false positive rate at different classification thresholds. The Area Under the ROC Curve (AUC) measures the area under the ROC curve. AUC ranges between 0 and 1. A model whose predictions are 100% wrong has an AUC of 0; one whose predictions are 100% correct has an An AUC of 1 represents a model with totally correct predictions, AUC of 0 indicate a model whose predictions are totally wrong.

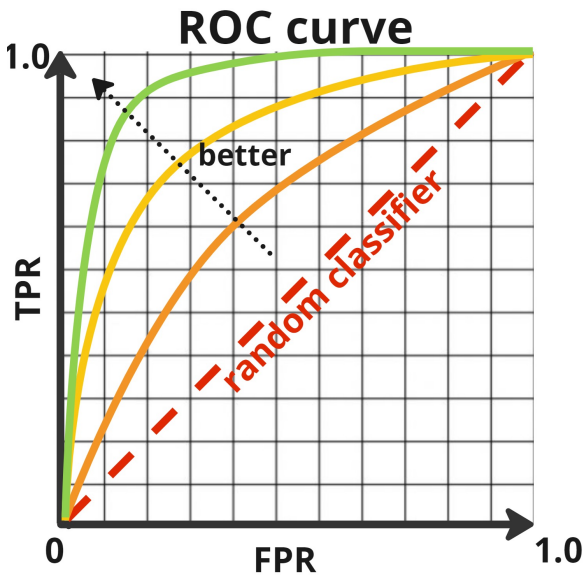


Figure 2.10: ROC curve explanation. As the curve goes up in the graph, the area under the curve increases and the performance of the classifier is better.

Another method to describe the performance of a classifier is through a **confusion matrix** (Figure 2.11). A confusion matrix for a binary classification problem includes four numbers:

- **True Positives (TP)**: The number of instances that are actually positive and correctly predicted as positive.
- **False Positives (FP)**: The number of instances that are actually negative but incorrectly predicted as positive.
- **True Negatives (TN)**: The number of instances that are actually negative and correctly predicted as negative.
- **False Negatives (FN)**: The number of instances that are actually positive but incorrectly predicted as negative.

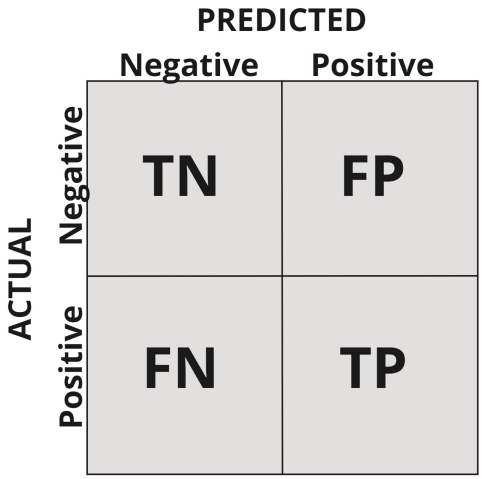


Figure 2.11: Confusion matrix explanation.

[36]. Overall, the choice of evaluation metrics for NN segmentation depends on the specific application and requirements of the task [37, 38].

2.3.3 Deep Segmentation and U-Net

Medical image segmentation is a challenging task that involves identifying the pixels of organs or lesions from background medical images like Magnetic Resonance (MR) or CT images. CNN have proven successful in this domain; however, traditional CNN require identical input shapes, which can be

problematic. To address this issue, the Fully Convolutional Network (FCN) was introduced in 2015 by Long et al [39]. The FCN has convolutional layers without fully connected layers, enabling it to predict arbitrary-sized inputs [40].

In the same year, the U-Net architecture was proposed by Ronnerberger et al. [41] to address the issue of loss of segmentation accuracy and insufficient integration of context information caused by the large multiplier used in the upsampling operation of FCN. The U-Net architecture uses an equal number of convolutional layers for upsampling and downsampling and incorporates skip connections between each level of the upsampling and downsampling layers. This allows features obtained from the downsampling layer to be passed to the upsampling layer, improving the accuracy of pixel positioning and segmentation [24].

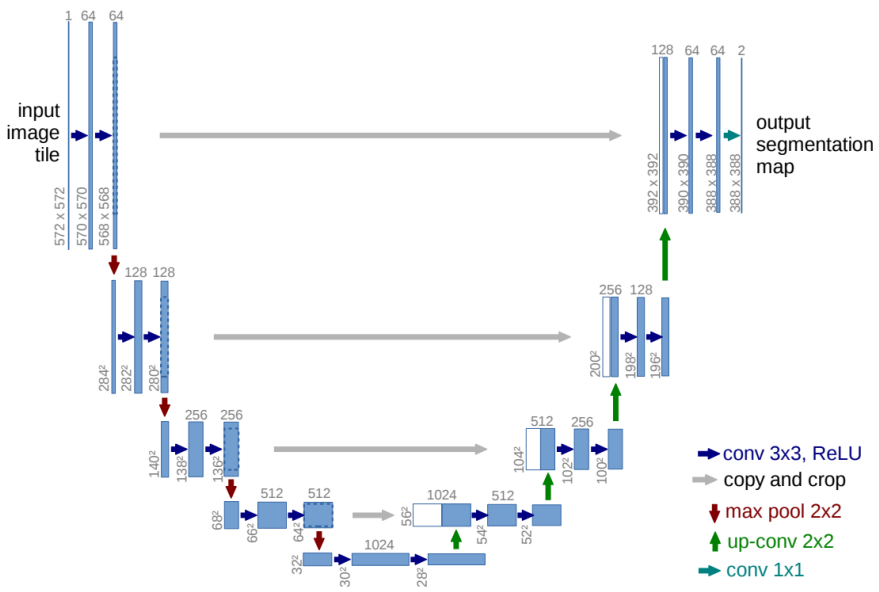


Figure 2.12: U-net architecture designed by Ronneberger et al. [41].

The architecture of the U-Net is illustrated in Figure 2.12 and comprises of a contracting path (also called analysis [42]) on the left side and an expansive path (also called synthesis) on the right side. The **contracting path** follows the typical structure of a convolutional network and is made up of repeated application of two 3x3 convolutions, each followed by a ReLu and a 2x2 Max Pooling (MaxPool) operation with stride 2 for downsampling. At every downsampling step, the number of feature channels is doubled. The **expansive path** involves an upsampling, i.e. an up convolution, of the feature map

that reduces the number of feature channels by half, a concatenation with the feature maps from the contracting path, and two 3x3 convolutions, each followed by a ReLu. The final layer of the network involves a 1x1 convolution to map the 64 components to the final number of output [41].

The architecture of the U-Net is fed with images and ground truth segmentation, also called masks, organized in a tensor structure. Tensors are multi-dimensional arrays that can have any number of dimensions and are the primary data structure used in deep learning frameworks like TensorFlow [43] and PyTorch [44]. A standard U-Net is trained using 2D images and therefore tensor of size $[C, H, W]$, where C represents the number of channels (for example 3 in RGB images), H is the height and W is the width of the image. For 3D images, also the depth D of the image is added as dimension.

The original U-Net model has undergone significant enhancements in segmentation networks, resulting in the creation of modern models that display remarkable performance in various challenging segmentation tasks, such as 3D U-Net [42], U-Net++ [45], nnU-Net [46], and others [47].

2.4 Probability and Uncertainty Mapping

Although automatic segmentation algorithms have achieved good accuracy in segmenting medical images, they still do not reach high levels of reliability in segmenting GTVs. This is likely due to the inter-observer variability inherent in the training data, the variability of imaging properties, and the heterogeneity of the GTV itself. To improve the robustness of these techniques, one promising approach is to incorporate uncertainty estimates of the automated segmentation results[8]. In medical image segmentation, uncertainty estimates reflect the confidence level of the predicted class label assigned to each voxel. A model with high confidence would exhibit low uncertainty, while a model with low confidence would have high uncertainty.

By utilizing uncertainty maps, it is possible to identify areas in the image where the segmentation is uncertain or ambiguous, which can be helpful for guiding manual corrections, pinpointing regions for further data analysis or collection, evaluating the quality of the segmentation, and detecting segmentation failures.

This uncertainty associated with the model prediction can be broken down into two main categories [48]:

- **aleatoric uncertainty**, due to noise within the data,
- **epistemic uncertainty**, due to the model architecture and parameters.

Aleatoric uncertainty can be further categorized into homoscedastic uncertainty, uncertainty which stays constant for different inputs, and heteroscedastic uncertainty which depends on the inputs to the model [49].

In literature, several methods for creating probability and uncertainty maps are presented, but three main categories can be delineated.
The first category is **Bayesian methods** which are based on the Bayesian inference framework. These methods estimate the posterior probability distribution of the parameter by using prior knowledge and observed data. The posterior distribution can be used to develop uncertainty maps. However, this method became intractable for a large number of parameters and therefore impossible for NN.

The second category is **Bayesian approximations** which use simplifications or approximations of the Bayesian inference framework to make the computations more tractable. Monte Carlo (MC) dropout presented by Gal and Ghahramani [50] in 2015 is one of the most popular methods in this category. As previously explained, dropout refers to the training procedure of randomly dropping a subset of nodes of a neural network in order to avoid overfitting and improve generalization ability [48]. Gal et al. suggested using dropout during the testing phase to produce N different segmentations that averaged together to build a probability map.

The third category is **non-Bayesian methods** which do not rely on the Bayesian inference framework and instead use other statistical or machine learning techniques to estimate uncertainty. Deep ensembles by Lakshminarayanan et al. [51] are the most famous, where results from multiple deterministic NNs trained with different parameter initializations are aggregated to determine the probability and uncertainty outputs.

In summary, each category of uncertainty mapping method has its own strengths and weaknesses, and the choice of method depends on the specific application, available data, and computational resources. Bayesian methods provide a principled framework for uncertainty estimation but can be computationally expensive, while Bayesian approximation methods provide a compromise between computational efficiency and accuracy. Non-Bayesian methods can be computationally efficient but may not provide accurate uncertainty estimates. Recently, new techniques for building such tools have been developed. One interesting attempt is represented by the probabilistic U-Net published by S. Kohl [52]. It builds upon the standard U-Net architecture but incorporates probabilistic modelling techniques. It predicts both segmentation masks and associated uncertainty maps, utilizing a variational autoencoder and a Bayesian approach.

2.4.1 Monte Carlo Dropout Technique

In 2015, Gal and Ghahramani developed a new theoretical framework where test-time dropout is demonstrated to be a good approximation of Bayesian inference, as shown in Figure 2.13. This framework offers a direct way to model uncertainty with dropout NN. This addresses the challenge of representing uncertainty in deep learning without compromising computational efficiency or accuracy [50].

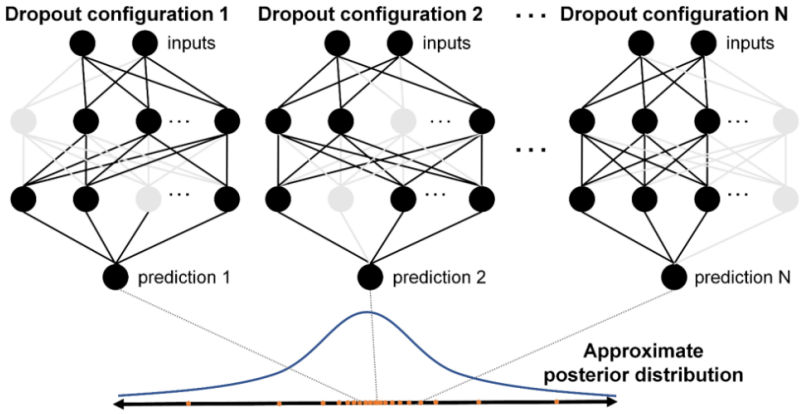


Figure 2.13: MC Dropout technique [53] (CC BY 4.0).

MC dropout is a method that does not require any prior information to be incorporated into the model and is capable of approximating the output distribution without any additional bias. Due to its similarity to Bayesian methods and straightforward implementation, MC dropout has become a popular choice for medical image analysis, as opposed to more complex Bayesian alternatives [48]. It is important to note that the utilization of MC dropout primarily predicts epistemic uncertainty, which helps capture the model's uncertainty in its own parameters. In 2020, Jungo et al. [8] analysed the effectiveness of MC dropout in the clinical context of automated brain tumour segmentation and confirmed the value of uncertainty estimation. In his paper, a detailed explanation of the procedure for obtaining probability and uncertainty maps is provided. First, dropout layers need to be included in the model, as shown in Figure 2.14, and the model trained.

At test-time, N random samples generated from the posterior distribution of the network's weights are considered as MC samples. The **foreground**

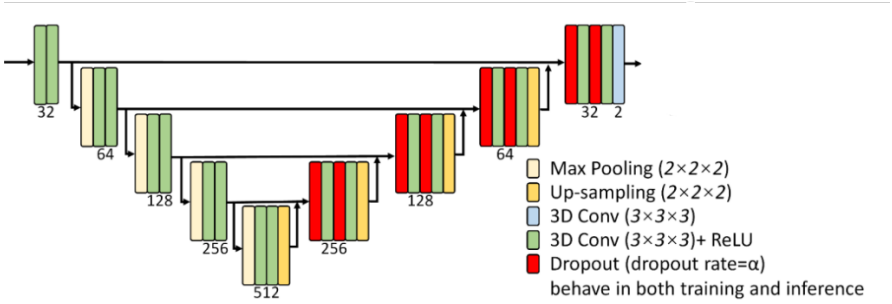


Figure 2.14: Dropout neural network [53] (CC BY 4.0).

probability is then computed by averaging the N samples using 2.8.

$$p_r = \frac{1}{T}\sum_{t=1}^{T} p_{r,t} \tag{2.8}$$

Normalized entropy (2.9) is then used as a measure of uncertainty.

$$\mathrm{H} = -[p_r \log p_r + (1-p_r)\log(1-p_r)]\frac{1}{\log 2} \in [0,1] \tag{2.9}$$

Different dropout strategies can be used. Jungo [8] describes in his article four different dropout methods used in a U-Net architecture. The first strategy involves using MC dropout in all layers with a **dropout rate** (likelihood of a neuron being switched off) of 0.05. The second method applies dropout only at specific key positions, for example at the centre of the architecture, while in the third one only at the two lowest pooling/upsampling steps. Finally, the fourth strategy uses concrete dropout, which learns the dropout probability during the optimization process.

Therefore, Jungo demonstrated the MC dropout method to be simple and easy to implement since it is only necessary to randomly switch off some nodes, create multiple segmentations and merge the results together. Moreover, this method involves less computational complexity than Bayesian NN and Ensembles, reduces over-fitting and improves generalization ability if applied also in the training phase. Also, as already stated, the method does not require any prior information compared to Bayesian NN, but it has still Bayesian-like outputs.

2.5 Model Reliability

Accurate uncertainty estimates are crucial for determining whether a model's output can be trusted and to ensure that the model's inference probabilities accurately reflect the likelihood of occurrence. Therefore, it is necessary to assess the quality of a model's reliability. It is also important to assess which methods are reliable under dataset shift, a common issue in medical data where changes occur between training, testing, and clinical distributions. One way to evaluate a model's confidence is by examining its calibration and comparing it to the perfect calibration. Calibration for segmentation models refers to the alignment between the model's predicted probabilities and the true probabilities of the predicted classes. Calibration is essential for reliable uncertainty estimation and ensuring that the model's confidence levels are meaningful. A model is considered perfectly calibrated when "*a model's prediction* $f(x)$ *with confidence* p *is correct with a rate of* p *for any label* y*, meaning that*:

$$P(y(x) = y|f(x) = p) = p, \tag{2.10}$$

where $y(x)$ *are the model's label predictions*" [8]. For example, if the confidence of a model is around 80%, it should give correct predictions 80 out of 100 times. A common graphical way to represent calibration is a reliability diagram. It plots the predicted confidence in bins against the observed accuracy within each bin. A well-calibrated model would have the points closely aligned with the ideal line (y=x). To obtain the so-called reliability diagrams, shown in Figure 2.15, the model's continuous predictions $f(x)$ need to be divided into M confidence bins and plotted against the accuracies in those bins, with the identity line on the **reliability diagram** representing perfect calibration. A model is considered overconfident when the curve of

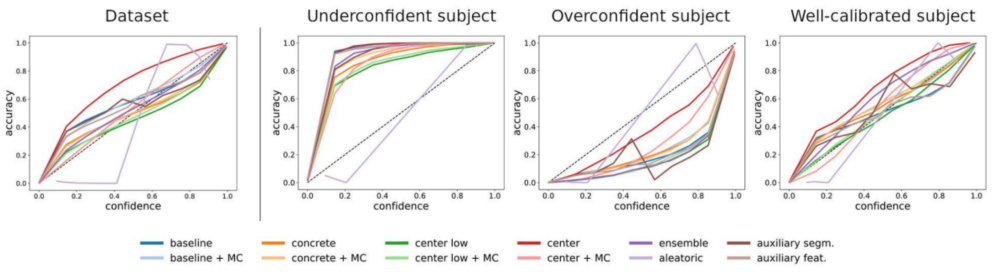


Figure 2.15: Comparison of different dropout strategies tried by Jungo for an underconfident, an overconfident and a well-calibrated subject (CC BY 4.0) [8].

its calibration plot falls below the identity line, and underconfident when it falls above. A popular way to quantify the model miscalibration using one scalar value is through the **Expected Calibration Error (ECE)**, given by the following equation:

$$ECE = \sum_{m=1}^{M} \frac{n_m}{N} |c_m - a_m|, \tag{2.11}$$

where c_m and a_m represent the confidence and the accuracy in bin m, M and N respectively the total number of bins and voxels, and n_m the number of voxels in bin m. ECE measures the discrepancy between predicted probabilities and observed accuracy. It divides the predicted probabilities into bins and calculates the average absolute difference between the average predicted probability and the observed accuracy within each bin. A lower ECE score indicates better calibration, with a value of 0 indicating perfect calibration.

Another common method to quantify reliability is to compute the miscalibration of uncertainty, called Expected Uncertainty Calibration Error (UCE) using the following equation:

$$UCE = \sum_{m=1}^{M} \frac{n_m}{N} |e_m - u_m|, \tag{2.12}$$

where e_m and u_m represent the error and the uncertainty in bin m, M and N respectively the total number of bins and voxels, and n_m the number of voxels in bin m [54]. Similar to ECE, UCE divides the predicted uncertainties into bins and calculates the average absolute difference between the average predicted uncertainty and the observed error rate within each bin. A lower UCE score indicates better calibration, implying that the model's predicted uncertainties align more closely with the actual errors made by the model.

Chapter 3

Methodology

This chapter outlines the methodology and technical choices that were made throughout the project and analyzes them in detail. The chapter begins with Section 3.1, which provides an overview of how the literature review was conducted. Section 3.2 focuses on the dataset used to train the NN and the pre-processing steps taken to clean and prepare the raw data. In Section 3.3, the implementation, training, and testing of a 3D U-Net for segmentation purposes are described. Section 3.4 explains the Monte Carlo Dropout technique, while Section 3.5 presents information on how to construct reliability diagrams for checking the model calibration. Finally, Section 3.6 delves into the clinical validation process and describes the experiment conducted to compare the model's results with the clinician's opinion. Figure 3.1 provides a visual representation of the project pipeline.

3.1 Research Process

The research process followed two different approaches in order to gather as much relevant information as possible.

The first approach started from scientific papers provided by the supervisor and was used as the foundation of the project [7, 8, 41]. Further papers were then researched based on the citation references. The articles obtained through this search were filtered based on their relevance. This first approach is useful to know the state of the art in this specific field of investigation and to check how far research has moved in that direction.

The second approach is more general and allows for a broader view of the research topic and a more structured search. A research question was

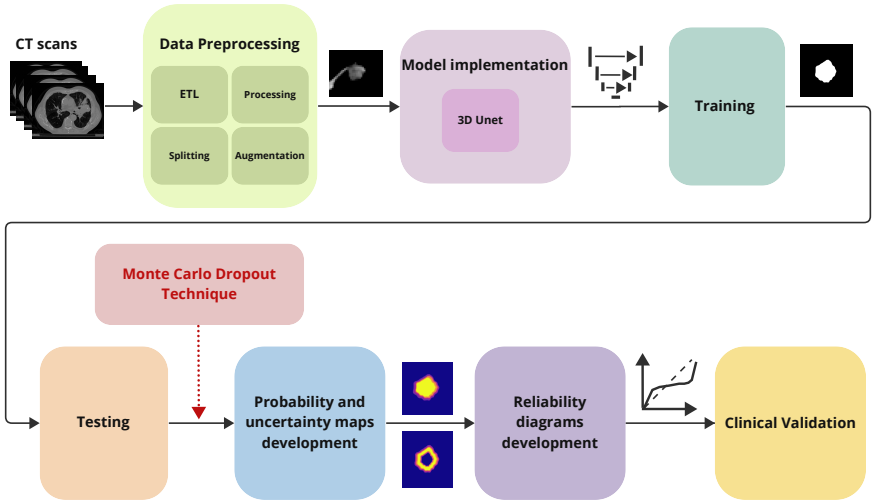


Figure 3.1: Project workflow. The Monte Carlo Dropout technique is applied at test time to produce probability and uncertainty maps.

formulated, search terms found and search blocks created. The formulated question is the following:

Does a probabilistic auto-contouring system using a 3D U-Net improve performances in clinical contouring practice for lung cancer CT images?

The defined keywords were used to search several databases by adopting search techniques such as truncation, phrase searching and Boolean operators (AND, OR). The databases used are Google Scholar, Web of Science Core Collection, PubMed and Scopus.

3.2 Data

In this section, the dataset used to train the NN will be illustrated. The different steps performed to process and augment the data will be described in detail as well.

3.2.1 Dataset

The 3D U-Net was trained on a collection of images from 422 NSCLC patients treated at MAASTRO Clinic, in The Netherlands [55, 56]. It is an open dataset published in the Cancer Imaging Archive. For each patient, they provided:

- pretreatment CT scans in the DICOM format

- manual contours by a radiation oncologist of the 3D volume of the primary gross tumour volume (GTV-1) and neighbouring OAR (i.e., right and left lung, heart, spinal cord and esophagus)

The dataset includes GTVs of various TNM stages since they vary in size and spread to nearby lymph nodes.

3.2.2 Extract, Load, and Transform Process

The dataset needed to be prepared through several preprocessing steps prior that such data being used for training and inference. An Extract Transform Load (ETL) process was performed to combine data from different files into a single central repository. The repository consists of a HDF5 file which allows for simpler and more structured handling. A data pre-processing pipeline was written and executed that extracted the Three-Dimensional (3D) image data from the DICOM file, patient-per-patient and stored it in the HDF5 file format as an array ([D, H, W], where D is depth, H is height and W is width). Along with the 3D image data, the relevant structure sets were extracted from the DICOM file, as binary images with the same coordinates of the images, defining a segmentation mask of the GTV that was manually delineated by the physicians. In addition, relevant metadata were extracted including:

- *rescale intercept* and *rescale slope* to convert pixel values to Hounsfield unit (HU) values;
- *slice thickness* to obtain the voxel spacing, hence the resolution in the z-direction;
- *pixel spacing* to get the resolution in the x-y direction.

CT scans and masks were saved as databases of the corresponding group, while metadata as attributes.

3.2.3 Data Preprocessing

After the ETL process, the processing of the data was performed. First, the dataset was cleaned up by removing patients with missing images and/or missing segmentation. For this project, the standard preprocessing steps for segmenting CT medical images were employed. The subsequent techniques outline the preprocessing procedures utilized:

- Voxel-space resampling: Process to obtain the same voxel size for each image in the dataset in order to make resolution homogeneous even though the patients were scanned with CT equipment from different manufacturers. In this project, the resampling size was fixed at *1x1x3 mm^3*.
- Transforming to HU: A linear transformation 3.1 applied to the data in order to get HU values from the image pixel values:

$$HU = PV * s + b \tag{3.1}$$

  where PV represents the pixel value, s is the rescale slope and b is the intercept.
- Contrast enhancement (HU windowing): HU values outside a predefined range {*min*, *max*} were truncated and all the values below *min* were set to *min* while all the values above *max* to *max*. A standard range for CT lung images is {*-1000*, *400*}, but other ranges were also tried.
- Intensity normalization: The HU values were normalized from the predefined range {*min*, *max*} to {*0*, *1*}.
- Cropping: Edges of the image are cut away. A cropping region of 1 cm, 3 cm and 5 cm were selected around the segmentation in order to keep the relevant neighbourhood for training and improve model training efficiency. These cropping approaches are acceptable as the final goal is probability and uncertainty, not the segmentation itself. The most performing cropping approach was then chosen.

Omitting the cropping technique is a possibility, but doing so would require feeding the model with an entire 3D image, which often results in memory constraints. However, it is important to note that all the other mentioned steps are indispensable for achieving satisfactory segmentation performance. Besides the application of cropping, other methods were tried out to deal with the memory issues arisen during the training. Downsampling represents a technique to reduce spatial resolution and consequently the storage size of the data. However, this technique may lead to throwing away relevant information. Another common technique is patching, i.e. subdividing the image into smaller patches. These patches can be treated as entire images, therefore given to the network and segmented. The final image can be reconstructed by merging together the segmentations of each patch.

To avoid the potential loss of relevant information from the images, denoising was not applied. Furthermore, since the images were confirmed to be aligned with the segmentations, there was no need for registration techniques to be utilized.

The final dataset was split into training, validation and testing sets following well-accepted general rules in the scientific community.

A flag was added as an attribute of each patient's group in the HDF5 file in order to distinguish the corresponding set to which each patient belongs, as illustrated in Table 3.1.

Set	Splitting	Flag
Training set	80% (333 patients)	”0”
Validation set	10% (42 patients)	”1”
Test set	10% (42 patients)	”2”

Table 3.1: Dataset split

In addition, data augmentation was applied at training time to increase the size of the training set using *the torchIO* Python library [57]. In particular, the following 4 transformations with a probability of 20% each were used:

- Flip along the x-axis: a technique that reverses rows and columns of the 3D matrix horizontally, i.e. along x-axis
- Flip along the y-axis: a technique that reverses rows and columns of the 3D matrix vertically, i.e. along the y-axis
- Flip along the z-axis: a technique that reverses rows and columns of the 3D matrix along the remaining axis, i.e. z-axis
- 90 degrees rotation: a technique that rotates the 3D image by 90 degrees clockwise from z- axis towards the x-axis.

Therefore, in 20% of cases, the data did not undergo any transformation.

3.3 Neural Network Model

In the following section, the details of the network architecture of the 3D U-Net and the related *Pytorch* implementation are provided. Moreover, both training and testing algorithms are described.

3.3.1 Model Architecture

The network architecture is illustrated in Figure 3.2 and summarised in Table 3.2. Like the standard U-Net by Ronneberger O. [41], this network architecture includes a contracting path and an expansive path, each with four resolution steps. The main difference consists in the replacement of 2D operations with 3D operations, making this network a 3D U-Net instead of a standard U-Net [42]. By using a 3D CNN, operations like convolution and pooling are implemented in a 3D space, preserving spatial information of the volumetric medical images [58].

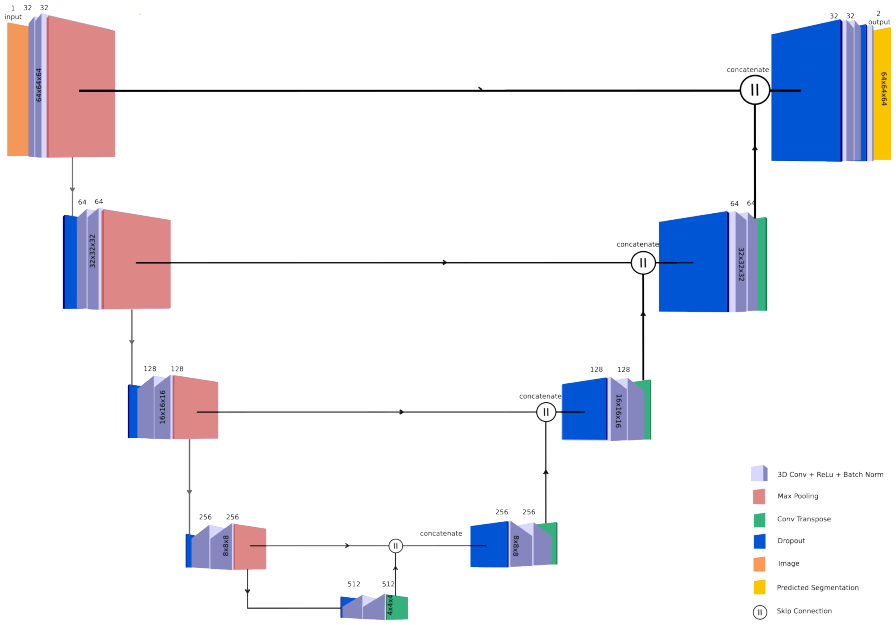


Figure 3.2: 3D U-Net architecture

The 3D U-Net was entirely built from scratch using open-source *PyTorch* packages but following literature parameters as reference. The primary motivation for this choice was driven by educational purposes. In the contracting path, each step includes two 3 x 3 x 3 3D Convolution (Conv) with padding of 1, each followed by a 3D Batch Normalization (BN) and ReLu. After this double convolution block, a 2 x 2 x 2 MaxPool operation and a dropout layer with *dropout rate (p)*= 0.2 are applied. A conservative approach was preferred when selecting a dropout rate in order to avoid dropping relevant information, as it is typically maintained within the range of 0.2 to 0.5. This step is repeated the same for the other three resolution steps of the contracting path. At each step of this path, the number of feature maps are doubled. In the

original U-Net model by Ronneberger O. (2015) [41], the number of feature maps that begin the network is 64, while the maximum number of feature channels is 1024. In this work, the minimum number has been chosen to be 32 and the maximum to 512, to reduce the memory issues that could arise during the training.

In the expansive path, each step consists of a 3D Transposed Convolution (ConvTransp) of 2 x 2 x 2, also called upconvolution, with a stride of 2, followed by a dropout layer ($p = 0.2$) and a double convolution with ReLu and BN. Skip connections from the contracting path are concatenated to feature maps of the expansive path according to their resolution step.

At the final step, another dropout layer and a 3D Conv are applied to map the 32 feature channels to the defined number of classes, i.e. 2.

As it can be noticed, dropout layers were added only at key positions [49] instead of being applied throughout all layers, as usually done to reduce overfitting [59]. In this project, dropout layers were used after each MaxPool operation in the contracting path and after each ConvTransp in the expansive path, plus another dropout layer at the final step.

3.3.2 Training and Testing

Two main parameters used during the training following the literature can be highlighted:

- Adam as the optimization method
- Dice Loss as loss function

The remaining hyperparameters were defined as a result of experimentation. Since a higher learning rate performs better on bigger batch size, batch size was set to 128 and learning rate to 1e-4 during the two approaches of training patches (32 x 32 x 32) and training images with the same size (80 x 112 x 32). The batch size was set to 1 and the learning rate to 1e-6 for the cropping approach, since a smaller learning rate performs better on small batch size [60].

A validation set was used to tune hyperparameters and avoid overfitting by plotting and comparing training and validation's loss function. Dice score was used as a metric to check model performance since it is a measure of overlap between the target and the prediction. The model was trained on an Nvidia Tesla T4 16GB GPU.

At testing time, dropout layers were switched off to predict segmentations. Softmax function 3.2 was used as an activation function at the final layer of

Blocks	Layers
Down 1	Conv3D+ReLu+BN Conv+ReLu+BN MaxPool+Dropout
Down 2	Conv3D+ReLu+BN Conv3D+ReLu+BN MaxPool+Dropout
Down 3	Conv3D+ReLu+BN Conv3D+ReLu+BN MaxPool+Dropout
Down 4	Conv3D+ReLu+BN Conv3D+ReLu+BN MaxPool+Dropout
Bottleneck	Conv3D+ReLu+BN Conv3D+ReLu+BN ConvTransp+Dropout
Up 1	Conv3D+ReLu+BN Conv3D+ReLu+BN ConvTransp+Dropout
Up 2	Conv3D+ReLu+BN Conv3D+ReLu+BN ConvTransp+Dropout
Up 3	Conv3D+ReLu+BN Conv3D+ReLu+BN ConvTransp+Dropout
Output	Conv3D+ReLu+BN Conv3D+ReLu+BN Dropout Conv3D+ReLu+BN

Table 3.2: Building blocks of the 3D U-Net architecture

the NN

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3.2}$$

Finally, the class with the highest probability was selected per each voxel as a predicted class label by using argmax function.

3.4 Monte Carlo Dropout

The first main goal of the project is to develop probability and uncertainty maps. Following Jungo's work on the topic [8], dropout layers have been used at test-time to create MC samples [7]. Therefore, dropout layers were kept on during the experiment and 25 predictions T were produced per each patient, following the procedure described in [54]. The foreground probability p_r related to the cancerous region r was computed as the average of the 25 MC samples using 2.8.

Probability maps were then obtained from the resulting tensor $[C, D, H, W]$ by extracting the second channel output of the tensor (output size: $[D, H, W]$). Normalized entropy 2.9 was employed as a measure of uncertainty [8], while the softmax output of the network was used a probability p_r of the GTV region r.

Softmax output, probability and uncertainty maps were saved into a HDF5 file for later use. Maps were visualized using *Image Slicer Viewer*, an open-source code using *Matplotlib* library [61].

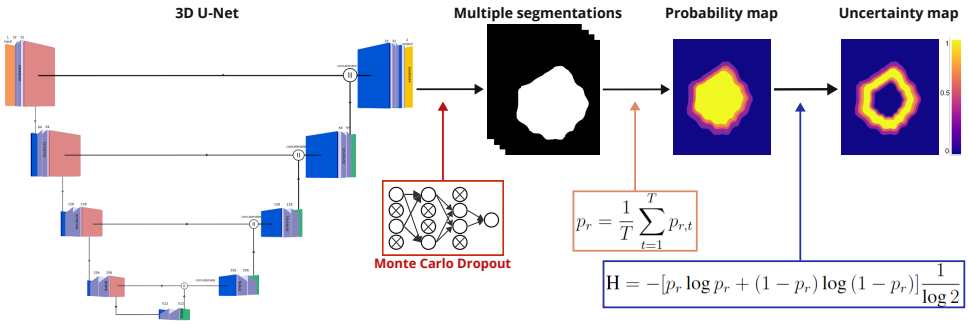


Figure 3.3: Pipeline to obtain probability and uncertainty maps using Monte Carlo Dropout technique.

3.5 Reliability Diagrams

The quality of the model's confidence was assessed by using reliability diagrams [62]. Two different reliability diagrams were produced. To create a reliability diagram which plots relative frequency as a function of confidence, predictions were discretized into 15 bins (M) with equal width, following the methodology delineated in [54] and [63]. Labels were considered as a measure of frequency, while the predicted softmax likelihood was a measure of

confidence. For each bin, voxels whose prediction confidence falls within that interval were considered. The weighted average of confidence and frequency within the bin were calculated and saved in a list. Once the results for each bin were obtained, a reliability diagram was plotted and the ECE was calculated. As for the second approach, a diagram was developed plotting the errors as a function of uncertainty, as shown in literature [54]. Uncertainty values, i.e., probability predictions converted into entropy 2.9, were averaged within each bin. Voxels with predictions different from the target that fell within that range were considered errors and averaged. Afterwards, UCE was computed and the reliability diagram was plotted. The selection of the aforementioned metrics was based on their established status as standard measures for evaluating the reliability of the model.

3.6 Clinical Validation

A clinical experiment was performed to evaluate the developed model, testing whether the results produced by the NN agreed with a clinician's opinion.

The test set used during the experiment was composed of 42 patients randomly selected from the dataset of 417 patients. These patients were not used during the previous training steps, constituting an unseen dataset. The experiment was conducted blind, as the clinician was shown only the CT scans and the target, but not the uncertainty maps produced by the model.

Two stages were included in the experiment:

- Uncertainty classification
- Uncertainty localization

The first phase involved the clinician's classification of each patient into three different categories:

- GTV with expected *high* segmentation uncertainty
- GTV with expected *intermediate* segmentation uncertainty
- GTV with expected *low* segmentation uncertainty

Dicompyler platform [64] was used to visualize the CT scans of each patient. The clinician was allowed to scroll through the slices, change the window width and level, zoom in and out, and visualize the overlapping GTV mask. Then, the clinician had to classify the patient into one of the three aforementioned categories.

During the second phase, the clinician had to indicate the area with the highest expected uncertainty in their opinion. Snapshots of the screen were captured while the clinician was indicating the areas with the pointer. In addition, the clinician's comments on each patient were noted. Once all the results were collected, the outputs were analysed by developing two metrics of evaluation:

- Mean Uncertainty (MU): average of the uncertainty values of each voxel in the uncertainty map,

$$\text{Mean Uncertainty} = \frac{1}{N}\sum_{n=1}^{N} H_n \tag{3.3}$$

  where N is the total number of voxels in the image and H_n represents the uncertainty value of the voxel n

- Relative Uncertainty Volume (RUV): volume of the higher uncertainty area,

$$\text{Relative Uncertainty Volume} = \frac{V_{\text{high uncertainty}}}{V_{\text{target}}} \tag{3.4}$$

Since it is known that the uncertainties scatter all over the images, it is necessary to narrow it to the high uncertainty regions. Otsu's thresholding technique was implemented to extract the high uncertainty volume [65]. This method was used to find the best uncertainty threshold in order to separate the voxels into two classes, the background and the ring related to the highest uncertainty area. The best threshold is represented by the uncertainty value that minimizes the intra-class intensity variance, i.e. the variability between observations of the same class. After finding the best threshold, the higher uncertainty class voxels were extracted and considered for computing the relative uncertainty volume.

In addition, ROC curves were plotted considering two different cases, representing the 2 opposite levels of clinician's concern. In the first case, patients classified as intermediate by the clinician were considered with low uncertainty. This depicts the scenario of a less concerned clinician, who wants to limit the use of this tool only to regions with very high uncertainty. In the second case, patients with intermediate uncertainty were added to the high uncertainty class. This depicts the approach of another clinician, more concerned, who prefers to include more regions in the high uncertainty area. In this way, it is possible to compare the performance of the two developed

metrics, MU and RUV, considering the two extreme cases. AUC score was computed as well to quantify the ability of discrimination of both metrics.

Furthermore, two sensitivity values were decided and used to classify the patients into two classes, low uncertain and high uncertain, using the two metrics. Confusion matrices were then employed to compare the different scenarios.

Chapter 4

Results

This chapter provides a comprehensive presentation of the results obtained throughout the thesis project. To facilitate parallel consideration of methodology and results, this chapter follows a similar section division to the previous one.
Section 4.1 outlines the results of the data handling and preprocessing, including details of the resultant images. In Section 4.2, the reader can find details regarding the performances of the 3D U-Net model previously described. MC Dropout technique was applied to generate probability and uncertainty maps, which are presented in Section 4.3, while Section 4.4 contains reliability diagrams used to check the calibration of the model. Finally, Section 4.5 illustrates results from the experiment designed to clinically validate the model.

4.1 Data Preprocessing

As mentioned earlier, the dataset used in this study comprises GTV of various sizes, each corresponding to a different TNM stage. The volume distribution of the primary GTV for the dataset is shown in Figure 4.1. The distribution appears to be half-normally distributed, with a peak of values near zero and a tail that extends to around 800 cm^3.

Figure 4.2 depicts an example of the data before preprocessing. On the left side, a 2D slice of the 3D scan of patient '1' is displayed. The values in the image are still pixel values and not windowed HU values. On the right side, the corresponding GTV delineation is shown.

As mentioned in the methodology chapter, the dataset was cleaned up by removing patients with missing images and/or segmentation. As a result, 5

patients were removed, which is a small number compared to the initial dataset of 422 patients.

Various preprocessing steps were implemented, as described previously. Figure 4.3 displays the different cropping techniques used, with the margin around the GTV increasing from left to right in the image.

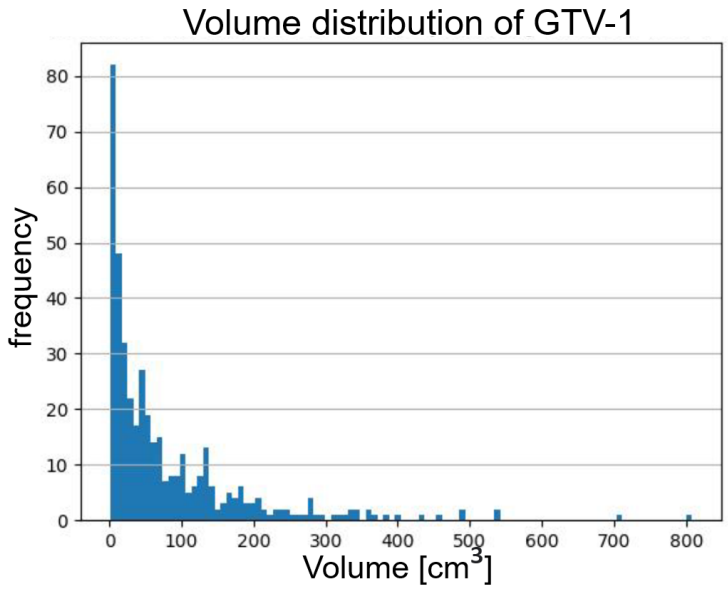


Figure 4.1: Volume distribution of the primary GTV in the dataset.

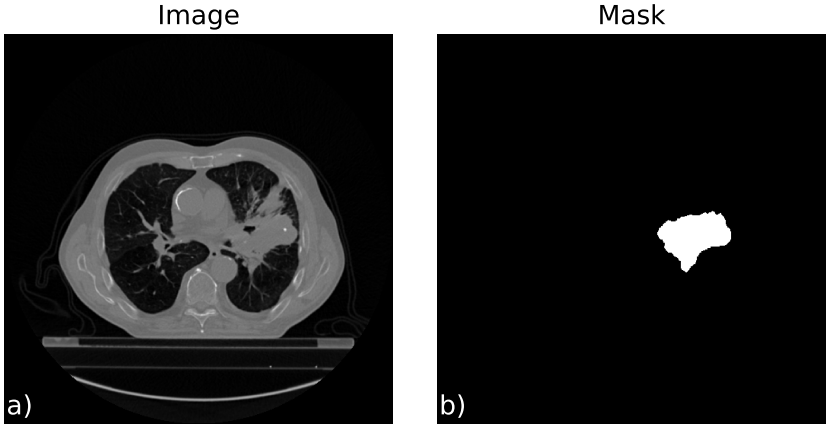


Figure 4.2: Example of a 2D slice of the CT image (*a*) and GTV delineations by a clinician (*b*) before preprocessing.

Several ranges of HU were experimented and the most effective one ([-

150, 250]) was selected. When the window width is decreased, the GTV's edges become more distinct and identifiable. However, it is crucial to select the appropriate window level to avoid losing critical information.
Figure 4.4 illustrates the difference between two images generated using different HU windows. Image *a* shows the result of the application of a HU window in the range [-1000, 400], while the second one corresponds to the same CT slide preprocessed with a HU window of [-150, 250].

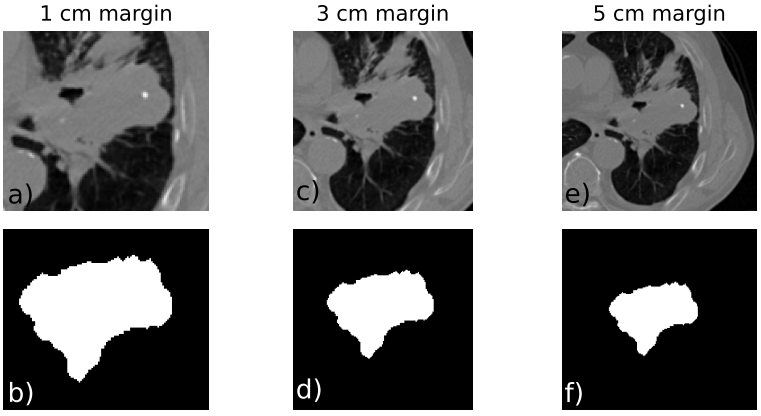


Figure 4.3: Image and corresponding mask after applying three different cropping sizes.

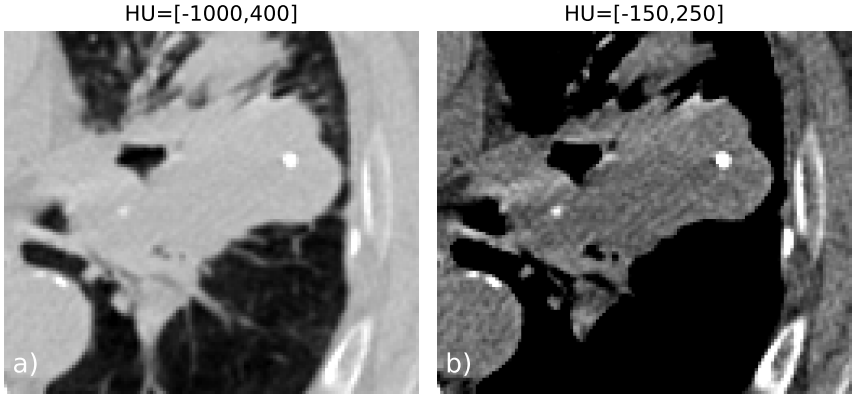


Figure 4.4: Comparison between two different HU windowing.

After resampling, windowing, and cropping, the preprocessed dataset underwent data augmentation techniques discussed in Section 3.2. Figure 4.5

demonstrates the effects of flipping a 2D slice dataset along all three axes.

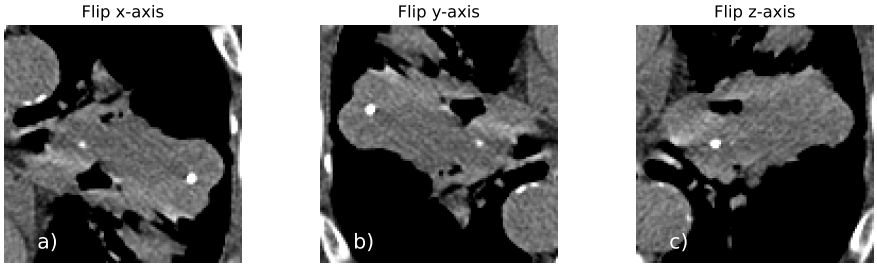


Figure 4.5: Data augmentation: flipping along the three axes.

4.2 Segmentation Performance

As mentioned in the preceding chapter, various preprocessing techniques were applied to the dataset during the training of the model to determine the most effective combination. The choice to use the dice score for evaluating the model is based on its effectiveness in measuring the similarity between the predicted segmentation and the ground truth. Moreover, it is commonly used in medical image analysis and computer vision tasks. The following three Figures, 4.6, 4.7 and 4.8, display performances in terms of the loss function and the dice score of the training of the datasets cropped with the three distinct methods. Results from the training set are represented by the blue curve, validation by the orange curve and a red line is used to indicate where overfitting starts occurring. Plot a shows the loss function of both training and validation, while plot b displays the dice score trend. Figure 4.6 demonstrates a clear instance of overfitting that arises after approximately 400 epochs, indicating that training should conclude around the 400th epoch. Although an early stopping technique could have been employed, it was not deemed essential to attain high segmentation performance for this project. As a result, through multiple experiments, it was consistently observed that a plateau was reached after approximately 400 epochs, leading to the selection of this number as the final training epoch count. Additionally, the same plot displays a gap between the training curve (blue line) and the validation curve (orange line), as expected due to overfitting.

Furthermore, it is noticeable that the model delivers better results when the dataset is cropped with only 1 cm of margin. A difference of about 0.20 in

the validation dice score is evident between the results with a margin of 1 cm and the two other cropping methods. As the margin increases, the model's performance appears to worsen. Hence, the first method was selected for subsequent phases. The average dice score achieved by the chosen model on the test set was 0.60 ± 0.25 (Table 4.1) Figure 4.9 illustrates an example of segmentation results generated by the model. Figure 4.9*a* represents a preprocessed CT image from the test set, figure 4.9*b* the corresponding delineation from the clinician used to train the model, figure 4.9*c* shows the prediction produced by the model and figure *d* the difference between the contour provided by the clinician and the one predicted. Results from the downsampling and patching methods are omitted since they did not produce relevant results.

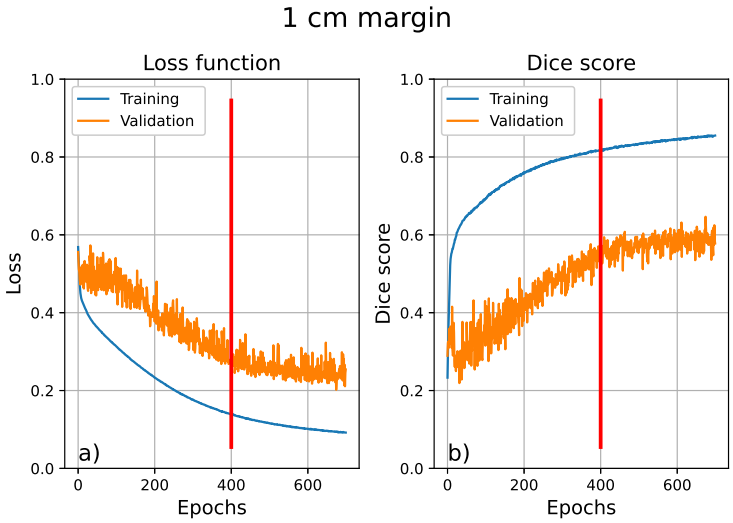


Figure 4.6: Model performance during training using the dataset cropped with 1 cm margin.

Training approach	Dice score
1 cm margin + HU=$[-150, 250]$	0.60 ± 0.25
1 cm margin + HU=$[-1000, 400]$	$0,55 \pm 0,28$
3 cm margin + HU=$[-150, 250]$	$0,50 \pm 0,31$
5 cm margin + HU=$[-150, 250]$	$0,44 \pm 0,26$

Table 4.1: Dice score of the main training approaches.

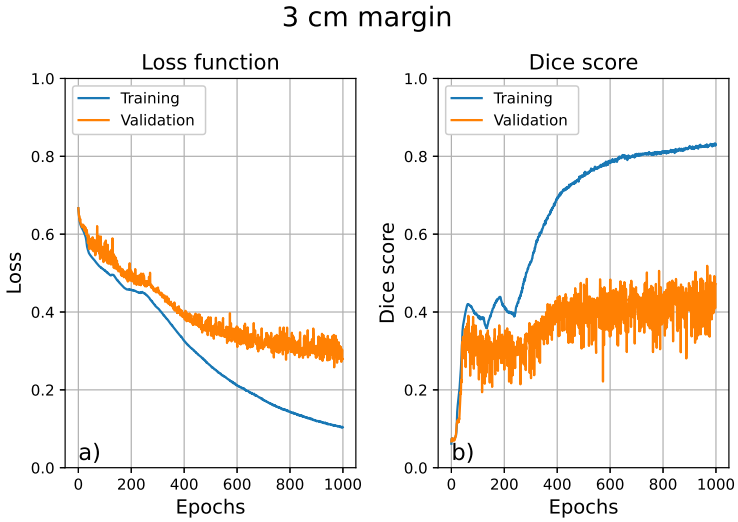


Figure 4.7: Model performance during training using the dataset cropped with 3 cm margin.

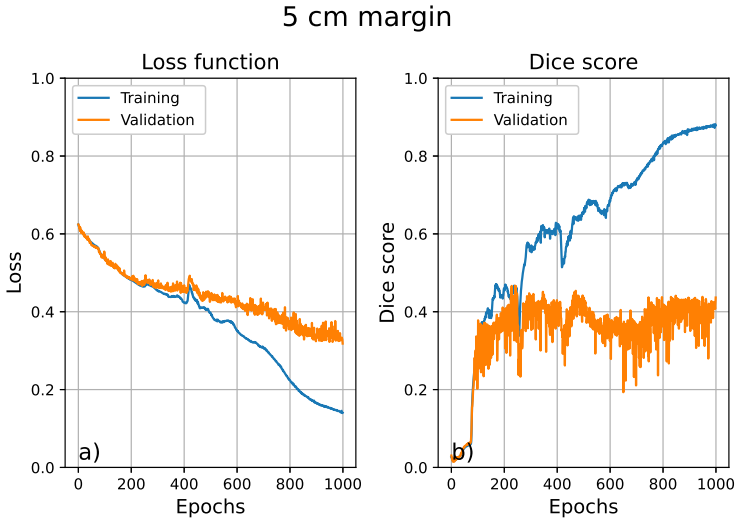


Figure 4.8: Model performance during training using the dataset cropped with 5 cm margin.

4.3 Probability and Uncertainty Mapping

Using the Monte Carlo dropout technique, as described in Chapter 3, the model generates probability and uncertainty maps. This section displays four different scenarios obtained from the outcomes. Each of the four figures

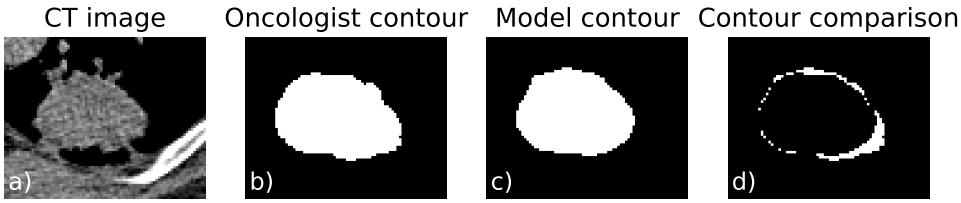


Figure 4.9: Example of segmentation results produced by the trained model.

consists of five images:

- (a) preprocessed CT image
- (b) manual delineation by a radiation oncologist
- (c) segmentation generated by the model
- (d) probability maps derived from the Monte Carlo dropout technique
- (e) uncertainty maps resulting from the application of the entropy equation 2.9

In the first case, depicted in Figure 4.10, it is observable that the oncologist did not contour an **ambiguous spike** connected to the spherical tumour area, which the model captured in its prediction. Nonetheless, the probability and uncertainty maps indicated low probability and high uncertainty in that region.

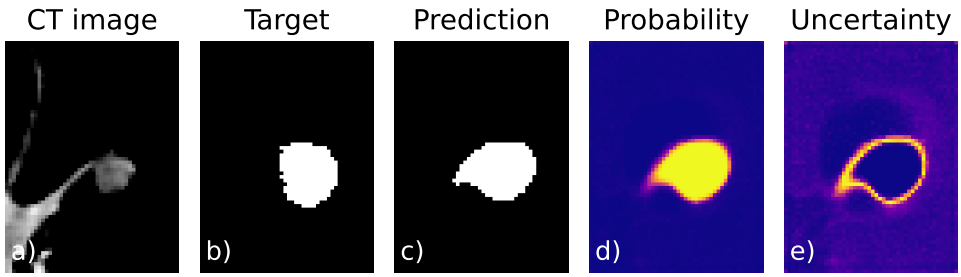


Figure 4.10: Example of probability and uncertainty maps revealing an ambiguous spike.

Figure 4.11 presents the second scenario, where the model's prediction appears to be **overly contoured** when compared to the target. Probability and uncertainty maps indicate again that area as low probable and highly uncertain.

Figure 4.12 displays a scenario where the model's contouring is smaller than the clinician's delineation, resulting in **under-contouring**. Two areas at

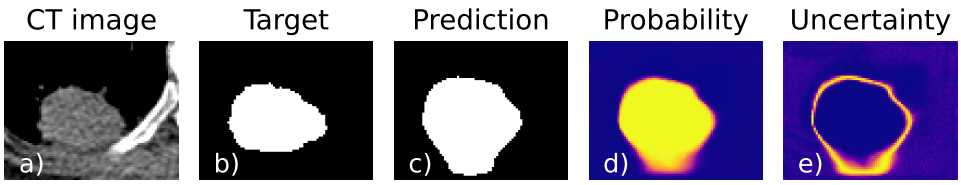


Figure 4.11: Example of probability and uncertainty maps revealing over-contouring.

the top and bottom of the contouring are left out in the prediction. Nonetheless, it is clear from images *d* and *e* of the same figure that probability and uncertainty maps detected the same shape contoured by the clinician, although the uncertainty is high.

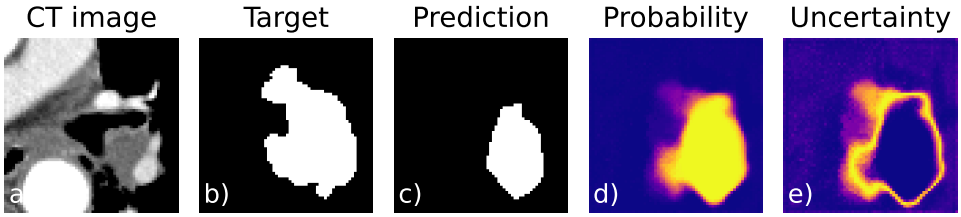


Figure 4.12: Example of probability and uncertainty maps revealing under-contouring.

The final example of **missed tumour** can be viewed as an extreme case of under-contouring. The model failed to provide any segmentation, resulting in the GTV not being recognized, even though the clinician marked a contour in that region. Nonetheless, both probability and uncertainty maps indicated the presence of a potentially cancerous area in that region.

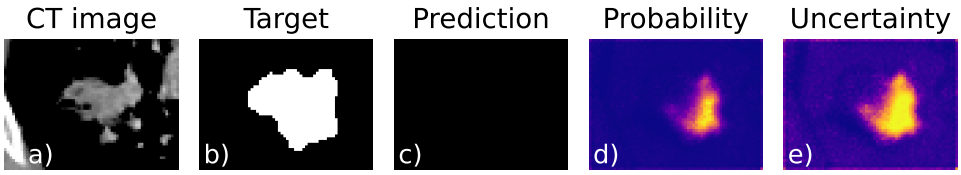


Figure 4.13: Example of probability and uncertainty maps revealing a missed cancerous area.

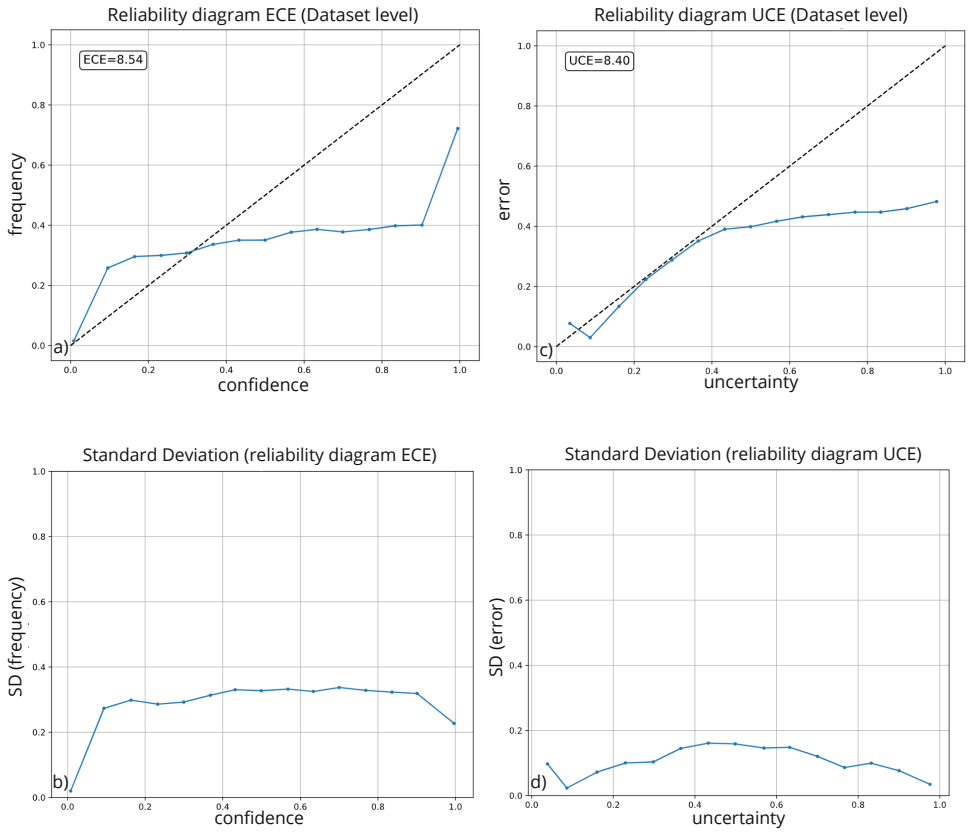


Figure 4.14: Reliability diagrams plotting relative frequency as a function of confidence (a) and error as a function of uncertainty (c) with corresponding standard deviation SD (b)(d).

4.4 Reliability Analysis

In Section 3.5, the methodology used to create the following reliability diagrams is explained. The results of this calibration check are presented in Figure 4.14, where four subplots are shown.

In Figure 4.14, plot 4.14a shows the averaged relative frequency plotted against the averaged confidence, while the plot 4.14b illustrates the standard deviation curve of the dataset frequency per each bin of the previous plot. Although the curve of the average does not precisely follow the identity line, it hovers around it.

In the same Figure, the mean error across all patients is plotted against the mean uncertainty in plot 4.14c, while image 4.14d shows the trend of the standard deviation related to dataset error. Once again, the curve of the

averaged error deviates from the ideal calibration line, especially for higher uncertainty values, revealing **overconfidence** in the model. The ECE and UCE scores defined in Section and computed as explained in Section 3.5 are displayed in the top left corner of the figures and in Table 4.2.

Metric	Score
ECE	8.54
UCE	8.40

Table 4.2: ECE and UCE scores obtained through reliability analysis

4.5 Clinical Validation

In this section, the results obtained from the clinical experiment are presented. To start with, an example of the uncertainty distribution of a patient is illustrated in Figure 4.15. As stated earlier, the bar plot displays two peaks that represent two distinct classes, the low uncertainty related to the background and the high uncertainty area related to the ring around the segmentation. The intra-class intensity variance is minimized to find the best threshold to separate the two classes. In the figure, this threshold is illustrated as a red line at almost the centre of the distribution. Figure 4.16 illustrate the outcomes of the application of Otsu thresholding to extract the high uncertainty volume and remove the background pixels with low uncertainty values.

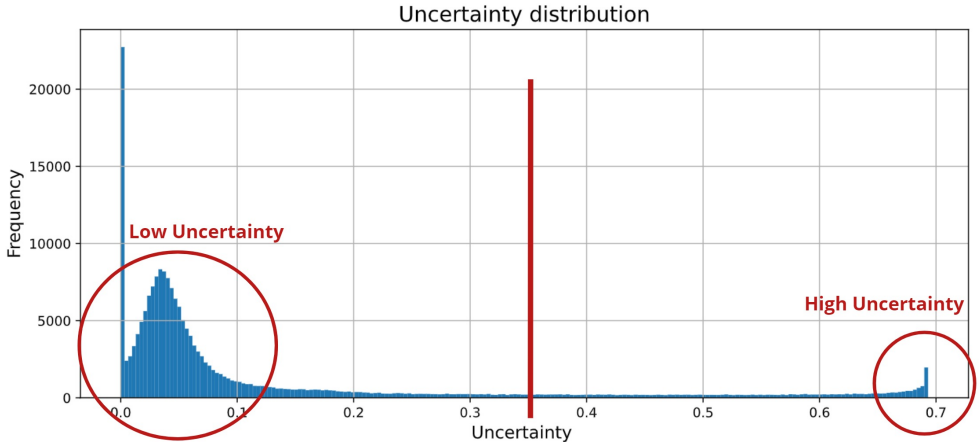


Figure 4.15: Example of uncertainty distribution of one patient of the test set.

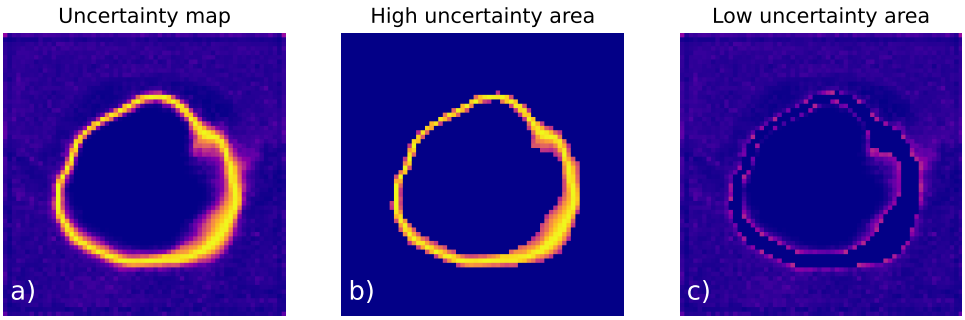


Figure 4.16: Application of Otsu thresholding on the uncertainty map (a) to extract high (b) and low(c) uncertainty areas.

Figure 4.17 shows the outcomes of the first metric, MU. This barplot presents the mean uncertainty values of the patients, with each bar coloured according to the clinician's feedback. Patients with low uncertainty are represented by blue bars, while high-uncertainty patients are shown in red, and those with intermediate uncertainty are depicted in yellow. A colour pattern can be observed, with blue bars dominating the left side of the figure, and red bars prevailing on the right. Yellow bars prevail in the middle but outliers can be found also in the red and blue areas.

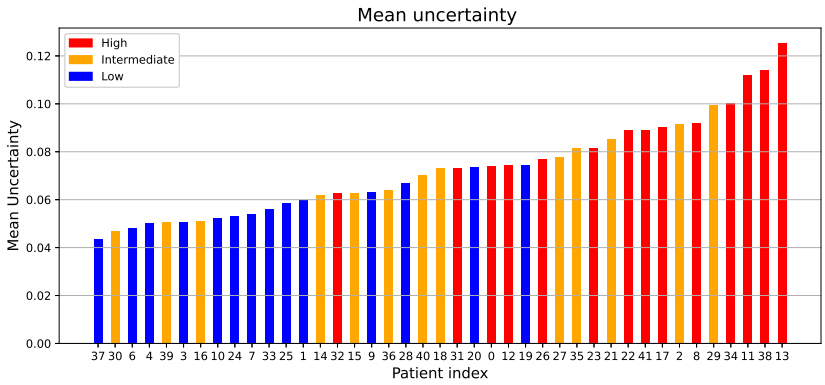


Figure 4.17: Bar chart of the correlation between the clinician's opinion (colour-coded) and the mean uncertainty score MU (y-axis) provided by the model per each patient.

The results of the second metric, RUV, are displayed in Figure 4.18, where the colors of the bars also follow a certain pattern but are less distinct than in the previous metric. In the appendix, Table A.1 contains all the results from the clinical experiment.

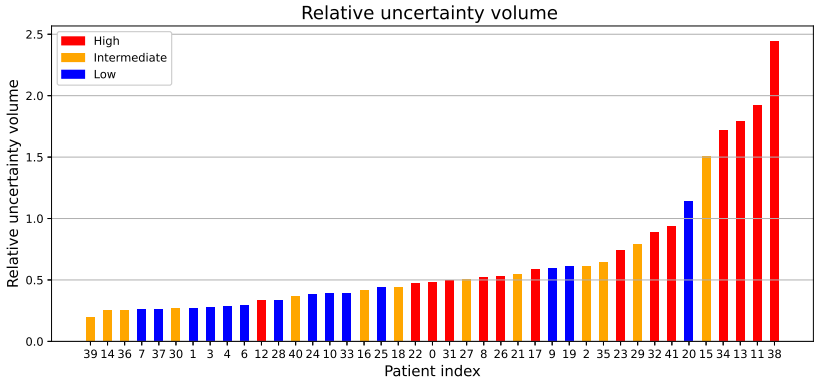


Figure 4.18: Bar chart of the correlation between the clinician's opinion (colour-coded) and the relative uncertainty volume RUV (y-axis) provided by the model per each patient.

Metric	Correlation coefficient
MU	0.68
RUV	0.49

Table 4.3: Pearson Correlation coefficient between metrics and clinical results

As presented in Table 4.3, one can notice that the MU has a higher Pearson correlation coefficient (a measure of the linear correlation between two variables) with the clinical outcomes compared to RUV metric. Figure 4.19 is a scatter plot that compares MU and RUV to check for any clusters related to the three classes. Blue dots accumulate in the lower-left corner with low values of both metrics, while red dots are mostly located in the upper area of the graph. Yellow dots are scattered in between, without a clear area of distinction. In the scatter plot, it is possible to notice that two dots are indicated with the corresponding number of the patient. For patient 12 it is possible to notice that the value of RUV is quite low, instead, the MU score is higher and seems to better discriminate this case as a high uncertain GTV. Patient 15 has a quite high value of RUV, while MU score is low and does not allow to gain the same classification as the clinician.

The ROC curves for both metrics are shown in figure 4.20. The two opposite levels of concern corresponding to two clinical cases of a less or more concerned clinician are illustrated in subplots 4.20*a* and 4.20*b*. In the first plot, intermediate patients were considered as having low uncertainty (image 4.20*a*), and in the second one as having high uncertainty (image *b*). It can be observed that MU outperforms RUV in both cases, with all ROC curves

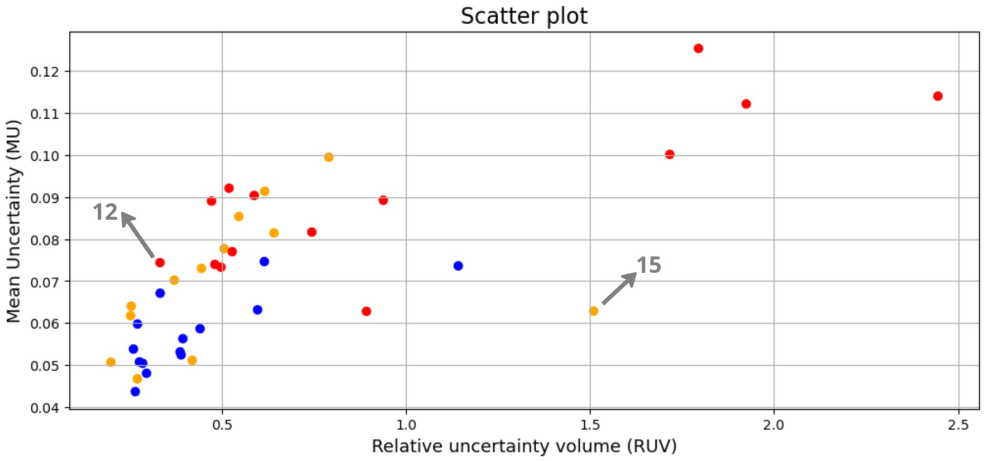


Figure 4.19: Scatter plot of the two validation metrics, MU on the y-axis and RUV on the x-axis. Two cases corresponding to patients 12 and 15 are marked.

having high AUC values, reaching a peak of 0.876 for MU in image 4.20*a*. AUC scores are summarized in Table 4.4, where the first rows show the results of the scenario of a less concerned clinician, and the last two rows the scenario of a more concerned clinician.

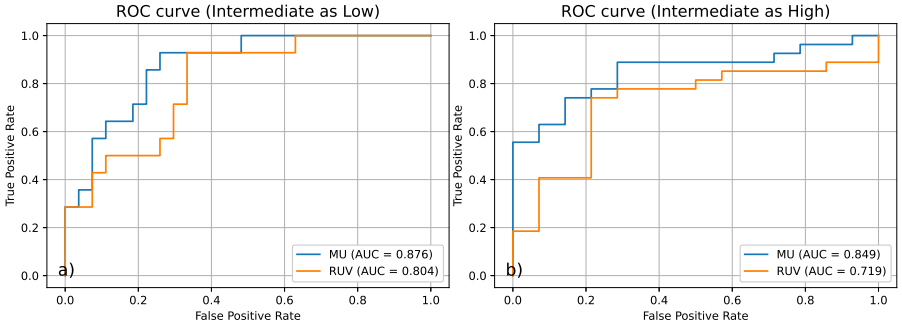


Figure 4.20: ROC curves of the two validation metrics, mean uncertainty (MU) and relative uncertainty volume (RUV) in two cases of sensibility.

The classification results after the definition of the two sensitivity levels (0.8 and 0.9) are summarized in Figures 4.21 and 4.22, where eight confusion matrices are illustrated. The first four matrices refer to the scenario of a less concerned clinician, while the remaining four to the case of a more concerned clinician. In each figure, the results from both metrics and both levels of sensitivity are shown.

Metric	AUC score
MU (less concern)	0.876
RUV (less concern)	0.804
MU (more concern)	0.849
RUV (more concern)	0.719

Table 4.4: AUC scores for MU and RUV considering the two approaches (less and more concerned clinicians).

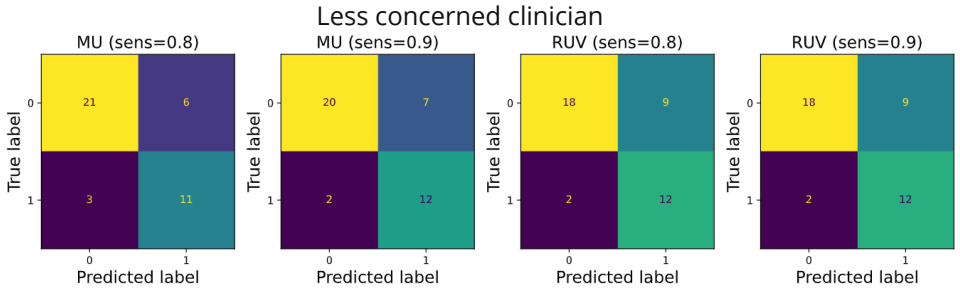


Figure 4.21: Confusion matrices: less concerned clinician (intermediate cases considered as low). Two levels of sensitivity (0.8 and 0.9) are displayed for each metric (MU and RUV).

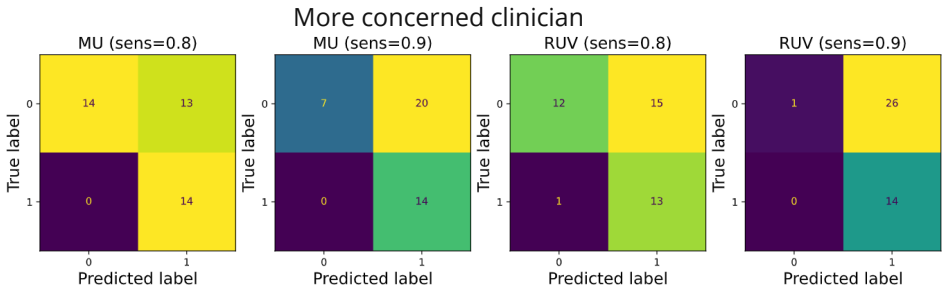
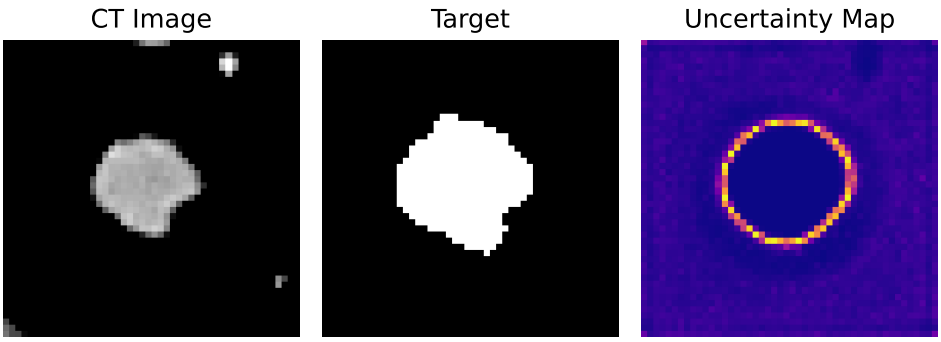
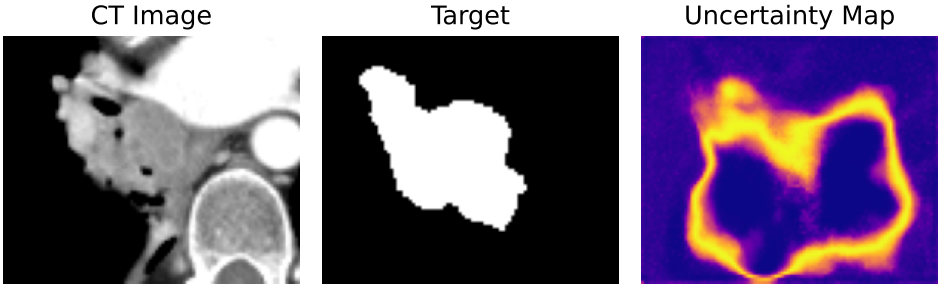


Figure 4.22: Confusion matrices: more concerned clinician (Intermediate cases considered as high). Two levels of sensitivity (0.8 and 0.9) are displayed for each metric (MU and RUV).

In the following Figures 4.23, two examples are extracted from the test set. Each of the two figure includes a CT image, the corresponding target delineated by a radiologist and the uncertainty map produced by model. The first case (4.23a) shows an example of GTV defined as low uncertain by the clinician. The uncertainty map seems to confirm the hypothesis producing a very thin and low uncertain ring around it. The second one shows an examples of GTV defined by the clinician as high uncertain. The uncertainty map seems



(a) 2D slice of a GTV (patient 4) defined as *low uncertain* by the clinician.

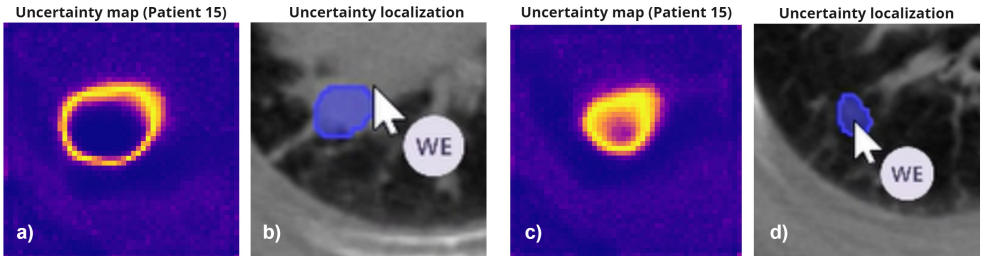


(b) 2D slice of a GTV (patient 38) defined as *high uncertain* by the clinician.

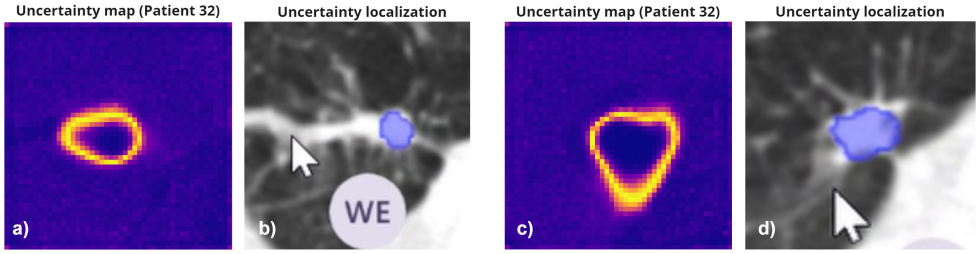
Figure 4.23: Examples of low and high uncertain GTV.

to agree with the clinician's opinion since the high uncertainty area seems to be quite large and thick.

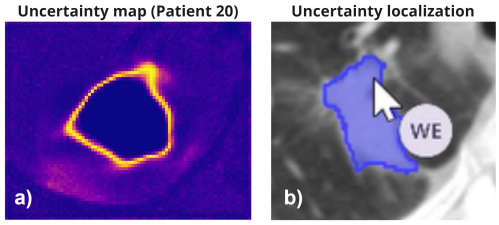
Figure 4.24 displays some outcomes obtained during the uncertainty localization phase of the experiment. For patient 15 (4.24a) and 32 (4.24b), two screenshots were captured during the experiment while the clinician was indicating with the pointer the region of high uncertainty in their belief. For patient 20 (4.24c), only one snapshot was captured. One can notice that a general agreement can be found between the indicated area and the thicker and brighter area of the uncertainty map.



(a) Patient 15.



(b) Patient 32.



(c) Patient 20.

Figure 4.24: Example of agreement between clinician's localization of high uncertainty areas and results from model's uncertainty maps.

Chapter 5

Discussion

This chapter analyzes and discusses the results presented in Chapter 4. The focus is mainly on the model's ability to generate probability and uncertainty maps and the clinical experiment.

First of all, despite building the model from scratch and not utilizing pre-built or pre-trained models, the segmentation performance for this dataset appears to agree with the literature for this dataset [9]. This performance may be limited due to the small number of patients and the large variability in GTV morphology. The latter was also reflected in the various TNM stages including in the patient list. Furthermore, a noticeable decline in model performance is evident in Figures 4.6 4.7 4.8, suggesting that the reduction of non-relevant information from the background may aid in improving the model's performance.

The results in terms of **probability and uncertainty maps** agree with observations in the literature. As in Wickstrom's study [66], the model tends to be confident in most of the voxels of the predictions, but it struggles to find defined borders of the GTV. This is the reason why the shape of the uncertainty maps is usually a ring that follows the margins of the GTV. It is difficult for the model to understand exactly where the cancerous area ends and the healthy tissue starts, as it is for a human. The probability maps are also reasonable since the probability tends to be higher within the target area and lower outside. By using a highly variable dataset, in which numerous different clinicians contoured the GTVs according to their own choices, the resulting uncertainty captures the inter-observer variability in the outcomes.

Section 4.3 presents four interesting cases that exemplify scenarios commonly observed in clinical practice. The first case (Figure 4.10) demonstrates the difficulty of defining borders of a GTV close to adjacent

vessels or other types of spikes. The ground truth of this GTV does not include anything related to the neighbouring spike, while the model prediction does include it. Probability and uncertainty maps highlight that area as low probable and highly uncertain, thereby warning the clinician to pay attention and double-check it. The clinician defined the same area as *"doubtful"*, thus confirming what was obtained from the model.

In the second figure (Figure 4.11), a scenario of over-contouring is represented. This example highlights how hard is for both clinicians and models to contour areas close to the mediastinum, diaphragm, or other anatomical structures. It is difficult to understand the exact pixels that mark the end of the GTV, especially because CT images have low contrast. In this case, the ground truth did not include part of the lower anatomical structure, while the model's prediction did. The uncertainty map revealed high uncertainty in that region, reflecting the struggle faced by clinicians in contouring in such conditions.

Figure 4.12 outlines an opposite scenario: under-contouring of the model's prediction compared to the ground truth. Here, it is clear that the clinician made choices related to the final aim of RT. The model did not include any of the surrounding tissues, resulting in under-contouring compared to the target. Both probability and uncertainty maps captured well the missed area.

In the final example presented to the reader in Figure 4.13, the model did not provide any prediction. This could result in a false negative case that will not be treated. However, both probability and uncertainty maps indicated a potential cancerous region.

All four previous examples indicate the usefulness of these maps in clinical practice to avoid cases of under or over-contouring that can lead to mistreatment, as well as to capture false negative cases in which no treatment would have taken place otherwise.

Calibration plots were used to assess the **reliability** of the model. The results in Figure 4.14 confirmed that the model is overconfident in the high probability area, which is a common problem for NNs [67]. The most likely reason for this issue is the overfitting and therefore the small dataset since it causes the learner to exhibit more confidence in its predictions that do not accurately reflect the actual data [68, 69]. This overconfidence is reflected in test data as well [70]. In this case, the model appears to be underconfident for low confidence bins, but mostly overconfident for values of confidence larger than 0.3, indicating miscalibration at the dataset level. However, the ECE and UCE scores (ECE = 8.54, UCE = 8.40) are lower compared to other uncalibrated models presented in [54], with scores reaching more than 30.

The standard deviation depicted in Figure 4.14 demonstrates higher values for intermediate confidence and uncertainty, with decreasing values as we move towards the extremes of the curve. Attempts to calibrate the model were made. The application of Temperature Scaling [54], a post-processing technique aimed at optimizing a scalar value called temperature to calibrate the softmax output at a global level, did not lead to successful results. This may be because local spatial miscalibrations occur in the images, and a global factor cannot fix this issue. Local scaling calibration approaches that address this phenomenon exist [71], but they were not implemented in this project. By applying local scaling calibration techniques, the model's output probabilities for each pixel or region in the segmented image can be adjusted or scaled, therefore at a local level.

Regarding Otsu's thresholding approach used to extract the higher uncertainty area and define the RUV metric, the results reveal consistency among the patients. The thresholds computed lay in the narrow range $[0.26, 0.29]$, thus confirming the reliability of the approach.

The final outcomes of the **clinical experiment** demonstrate a general agreement between the clinician's opinion and the model results. The barplots in Figures 4.17 and 4.18 reveal a pattern, which is supported by the Pearson correlation coefficient between the colour labels and the metrics scores ($r_{MU} = 0.68, r_{RUV} = 0.49$).

The two ROC curves, employed to reproduce the two different levels of concern that a clinician can express, exhibit high AUC values, particularly for the MU metric in the less concerned approach ($AUC_{MU} = 0.876$). The less concerned level, which considers intermediate cases as having low uncertainty (Figure 2.10 b), appears to perform better since the AUC scores are higher compared to the more concerned case (Figure 2.10 b). Decreasing the concern, which reduces the number of highly uncertain cases, produces a higher true positive rate and a lower false positive rate. One possible explanation for these results is that lowering these intermediate cases helps alleviate the model's overconfidence, which tends by its nature to lead to overestimated uncertainty. This is also evident in the confusion matrices (Figure 2.11), where the less concerned approach shows higher values of true positives and true negatives, with a minimal number of incorrect predictions. However, the more concerned approach allows for eliminating false negatives almost entirely.

The selected example figures from the test set further confirm the consistency achieved in the initial phase of the experiment, the uncertainty classification. In particular, Figures 4.23 highlight the distinctions between low and high uncertainty cases in the uncertainty maps, showing the model's

discriminative capability. It can be noticed that low uncertainty GTVs are typically characterized by low MU scores and a thin ring around the GTV, while high uncertainty cases exhibit thicker rings and brighter pixels in the uncertainty maps. Therefore, the two defined metrics appear to discriminate well these characteristics, as confirmed by the clinician. However, MU shows better performance as a classifier compared to the other metric. Therefore, the integration of uncertainty maps and their corresponding MU score in clinical practice could be employed as an indicator for clinicians to determine whether correction of the predictions is necessary or not. However, more research is still needed to incorporate uncertainty maps into clinical practice.

Figure 4.23a displays a GTV slice from patient 4, which the clinician described as having *"a peripheral location with little tissue around it, and not much proximity to blood vessels"*. Therefore, the clinician concludes with *"Not too much to expect here"*. Both MU and RUV metrics yield very low scores, thus confirming the clinician's observation. Figure 4.23b shows a case of very high uncertainty due to "*areas of fluid or puss similar to tumour*" close to the GTV itself, as stated by the clinician and found in the uncertainty maps.

The uncertainty localization phase of the clinical validation provided valuable insights, confirming the match between highly uncertain regions identified by the physician and those captured by the model. The clinician noted that areas closer to the diaphragm and mediastinum present more uncertainty and that spikes and adjacent vessels make it difficult to obtain well-defined segmentation boundaries. This aligns with the model's output, as demonstrated by slices from patients 15, 32, and 20 in Figure 4.24, where the clinician identified the same highly uncertain areas outlined in the uncertainty maps. However, not all patients in the test set exhibit such agreement, as evidenced by the two bar plots (Figures 4.17 and 4.18), which reveal also instances of colour mismatch.

Chapter 6

Conclusions

In the field of medical image segmentation, the high inter-observer variability poses a major challenge that leads to inaccurate segmentations and represents the primary source of errors in contouring tasks. The objective of this project was to investigate whether the use of a 3D U-Net probabilistic auto-contouring system could improve clinical contouring practice. To achieve this, a well-established technique was applied to generate probability and uncertainty maps, with the aim of demonstrating their usefulness as supplementary tools for clinicians in assessing segmentation accuracy. The methodology involved several steps to achieve this goal, which included building a 3D U-Net, implementing the Monte Carlo dropout technique, conducting reliability assessments, and finally clinically validating the model. The 3D U-Net achieved a dice score consistent with the performance of other works using the same architecture and dataset. The model produced reasonable probability and uncertainty maps by applying the Monte Carlo dropout technique, which captured interesting cases such as under-contouring, over-contouring or missed tumours. Reliability diagrams employed for checking the model calibration revealed overconfidence in the model. Despite this, the clinical validation performed with a radiation oncologist provided evidence of the effectiveness of the model, demonstrating a high level of agreement between the model results and the radiation oncologist's opinion. Two metrics were defined to quantify the uncertainty map's content. The experiment revealed a high correlation between the two metrics and the clinician's classification. However, MU outperformed RUV as a potential indicator for clinicians to determine whether correction of the predictions is necessary. Overall, the application of the Monte Carlo dropout technique resulted in valuable outcomes in terms of probability and uncertainty mapping. The clinical

validation results showed promising potential for probabilistic models to be a tool in clinical practice that can assist clinicians in their contouring tasks, resulting in significant time savings and a potential decrease in clinical errors. Uncertainty maps provide further information to the RO, which is not captured in the predictions. This allows for a more efficient correction process for the clinicians since it can help to recognize cases that could potentially lead to mistreatment, recurrence of cancer, or toxicity issues. By adjusting the sensitivity level of the uncertainty, a clinician can include more or less tissue within the target based on the specific clinical case and the aim of the segmentation. Additionally, thresholds can be established for the MU and RUV scores to activate an alert for the clinician when high levels of uncertainty are detected. In the case of low uncertainty predictions, the clinician could confidently proceed without spending additional time on double-checking the segmentation, as these predictions are automatically approved.

6.1 Future Work

The project demonstrates the potential of the Monte Carlo Dropout technique in producing probability and uncertainty maps that align with clinical opinions. However, further research is required to incorporate this tool into daily clinical practice effectively. Investigating the calibration of the model in greater detail would undoubtedly increase the tool's trustworthiness. Local scaling calibration could solve the problem of the local miscalibration of the model, therefore mitigating the overconfidence of the model and leading to more reliable outcomes. Furthermore, applying probability and uncertainty mapping to the segmentation of organs at risk, in conjunction with the GTV, could yield promising results. Additionally, the integration of other imaging techniques has the potential to improve the performance of this probabilistic auto-contouring system, effectively addressing the challenges associated with low contrast in CT imaging. Moreover, the integration of reinforcement learning of human expertise could enable the segmentation model to utilize domain knowledge, expert annotations, and feedback, thereby potentially enhancing the development of uncertainty maps. Lastly, to further validate the tool and enhance the value of the outcomes, a larger-scale experiment involving multiple clinicians is required.

References

[1] Y. Xia, Y. Zhang, F. Liu, W. Shen, and A. L. Yuille, "Synthesize then compare: Detecting failures and anomalies for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 145–161. [Page 1.]

[2] R. Challen, J. Denny, M. Pitt, L. Gompels, T. Edwards, and K. Tsaneva-Atanasova, "Artificial intelligence, bias and clinical safety," *BMJ Quality & Safety*, vol. 28, no. 3, pp. 231–237, 2019. [Page 1.]

[3] T. DeVries and G. W. Taylor, "Leveraging uncertainty estimates for predicting segmentation quality," *arXiv preprint arXiv:1807.00502*, 2018. [Pages 1 and 2.]

[4] A. Jungo, R. Meier, E. Ermis, E. Herrmann, and M. Reyes, "Uncertainty-driven sanity check: Application to postoperative brain tumor cavity segmentation," *arXiv preprint arXiv:1806.03106*, 2018. [Page 1.]

[5] A. Jungo, R. Meier, E. Ermis, M. Blatti-Moreno, E. Herrmann, R. Wiest, and M. Reyes, "On the effect of inter-observer variability for a reliable estimation of uncertainty of medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*. Springer, 2018, pp. 682–690. [Page 1.]

[6] K. Hoebel, V. Andrearczyk, A. Beers, J. Patel, K. Chang, A. Depeursinge, H. Müller, and J. Kalpathy-Cramer, "An exploration of uncertainty information for segmentation quality assessment," in *Medical Imaging 2020: Image Processing*, vol. 11313. SPIE, 2020, pp. 381–390. [Page 1.]

[7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international*

conference on machine learning. PMLR, 2016, pp. 1050–1059. [Pages 2, 31, and 39.]

[8] A. Jungo, F. Balsiger, and M. Reyes, "Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation," *Frontiers in neuroscience*, p. 282, 2020. [Pages ix, 2, 25, 27, 28, 29, 31, and 39.]

[9] S. Hossain, S. Najeeb, A. Shahriyar, Z. R. Abdullah, and M. Ariful Haque, "A pipeline for lung tumor detection and segmentation from ct scans using dilated convolutional neural networks," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. doi: 10.1109/ICASSP.2019.8683802 pp. 1348–1352. [Pages 4 and 59.]

[10] A. L. Oliver, "Lung cancer: epidemiology and screening," *Surgical Clinics*, vol. 102, no. 3, pp. 335–344, 2022. [Pages 7 and 8.]

[11] Cancer statistics, 2021. [Online]. Available: https://acsjournals.onlinelibrary.wiley.com/doi/10.3322/caac.21654 [Page 7.]

[12] N. Duma, R. Santana-Davila, and J. R. Molina, "Non–small cell lung cancer: epidemiology, screening, diagnosis, and treatment," in *Mayo Clinic Proceedings*, vol. 94, no. 8. Elsevier, 2019, pp. 1623–1640. [Pages 8 and 10.]

[13] J. R. Molina, P. Yang, S. D. Cassivi, S. E. Schild, and A. A. Adjei, "Non-small cell lung cancer: epidemiology, risk factors, treatment, and survivorship," in *Mayo clinic proceedings*, vol. 83, no. 5. Elsevier, 2008, pp. 584–594. [Pages 8 and 9.]

[14] What are the differences between small cell and non-small cell lung cancer? [Online]. Available: https://massivebio.com/what-are-the-differences-between-small-cell-and-non-small-cell-lung-cancer/#:~:text=What%20are%20the%20Differences%20Between%20SCLC%20and%20NSCLC%3F,spreads%20to%20the%20lymph%20nodes. [Page 8.]

[15] O. Lababede and M. A. Meziane, "The eighth edition of tnm staging of lung cancer: reference chart and diagrams," *The oncologist*, vol. 23, no. 7, pp. 844–848, 2018. [Page 9.]

[16] L. G. Collins, C. Haines, R. Perkel, and R. E. Enck, "Lung cancer: diagnosis and management," *American family physician*, vol. 75, no. 1, pp. 56–63, 2007. [Page 9.]

[17] C. Gridelli, A. Rossi, D. P. Carbone, J. Guarize, N. Karachaliou, T. Mok, F. Petrella, L. Spaggiari, and R. Rosell, "Non-small-cell lung cancer," *Nature reviews Disease primers*, vol. 1, no. 1, pp. 1–16, 2015. [Page 9.]

[18] Lung cancer diagnosis. [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/lung-cancer/lung-cancer-diagnosis#:~:text=Lung%20cancer%20is%20diagnosed%20through,emission%20tomography%20(PET)%20scans. [Page 9.]

[19] Radiation therapy for lung cancer. [Online]. Available: https://www.lung.org/lung-health-diseases/lung-disease-lookup/lung-cancer/treatment/types-of-treatment/radiation-therapy#:~:text=Lung%20cancer%20radiation%20therapy%20uses,radiation%20is%20used%20most%20often. [Page 10.]

[20] Chemotherapy for lung cancer. [Online]. Available: https://www.cancerresearchuk.org/about-cancer/lung-cancer/treatment/chemotherapy-treatment#:~:text=Chemotherapy%20is%20the%20main%20treatment,the%20bloodstream%20around%20the%20body. [Page 10.]

[21] J.-J. Sonke and J. Belderbos, "Adaptive radiotherapy for lung cancer," in *Seminars in radiation oncology*, vol. 20, no. 2. Elsevier, 2010, pp. 94–106. [Page 10.]

[22] G. Marvaso, M. Pepa, S. Volpe, F. Mastroleo, M. Zaffaroni, M. G. Vincini, G. Corrao, L. Bergamaschi, K. Mazzocco, G. Pravettoni *et al.*, "Virtual and augmented reality as a novel opportunity to unleash the power of radiotherapy in the digital era: A scoping review," *Applied Sciences*, vol. 12, no. 22, p. 11308, 2022. [Pages ix and 10.]

[23] M. P. Starmans, S. R. van der Voort, J. M. C. Tovar, J. F. Veenland, S. Klein, and W. J. Niessen, "Radiomics: data mining using quantitative medical image features," in *Handbook of medical image computing and computer assisted intervention*. Elsevier, 2020, pp. 429–456. [Page 12.]

[24] X. Liu, K.-W. Li, R. Yang, and L.-S. Geng, "Review of deep learning based automatic segmentation for lung cancer radiotherapy," *Frontiers in Oncology*, vol. 11, p. 717039, 2021. [Pages 12, 13, and 24.]

[25] N. G. Burnet, S. J. Thomas, K. E. Burton, and S. J. Jefferies, "Defining the tumour and target volumes for radiotherapy," *Cancer Imaging*, vol. 4, no. 2, p. 153, 2004. [Page 13.]

[26] S. P. Primakov, A. Ibrahim, J. E. van Timmeren, G. Wu, S. A. Keek, M. Beuque, R. W. Granzier, E. Lavrova, M. Scrivener, S. Sanduleanu *et al.*, "Automated detection and segmentation of non-small cell lung cancer computed tomography images," *Nature communications*, vol. 13, no. 1, p. 3423, 2022. [Page 13.]

[27] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021. [Pages 14 and 15.]

[28] H. Kukreja, N. Bharath, C. Siddesh, and S. Kuldeep, "An introduction to artificial neural network," *Int J Adv Res Innov Ideas Educ*, vol. 1, pp. 27–30, 2016. [Page 15.]

[29] Mplsvpn - moving towards sdn and nfv based networks: What is neuron and artificial neuron in deep learning? [Online]. Available: http://www.mplsvpn.info/2017/11/what-is-neuron-and-artificial-neuron-in.html [Pages ix and 16.]

[30] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015. [Page 18.]

[31] V. H. Phung and E. J. Rhee, "A deep learning approach for classification of cloud image patches on small datasets," *Journal of information and communication convergence engineering*, vol. 16, no. 3, pp. 173–178, 2018. [Pages ix and 18.]

[32] Batch normalization in convolutional neural networks. [Online]. Available: https://www.baeldung.com/cs/batch-normalization-cnn [Page 19.]

[33] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017. [Page 19.]

[34] About dicom: Overview. [Online]. Available: https://www.dicomstandard.org/about-home [Page 19.]

[35] The hdf5 library file format. [Online]. Available: https://www.hdfgroup.org/solutions/hdf5/ [Page 20.]

[36] P. Singh, N. Singh, K. K. Singh, and A. Singh, "Diagnosing of disease using machine learning," in *Machine learning and the internet of medical things in healthcare*. Elsevier, 2021, pp. 89–111. [Page 23.]

[37] M. Z. Khan, M. K. Gajendran, Y. Lee, and M. A. Khan, "Deep neural architectures for medical image semantic segmentation," *IEEE Access*, vol. 9, pp. 83 002–83 024, 2021. [Page 23.]

[38] P. Malhotra, S. Gupta, D. Koundal, A. Zaguia, and W. Enbeyle, "Deep neural networks for medical image segmentation," *Journal of Healthcare Engineering*, vol. 2022, 2022. [Page 23.]

[39] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440. [Page 24.]

[40] M. Aljabri and M. AlGhamdi, "A review on the use of deep learning for medical images segmentation," *Neurocomputing*, 2022. [Page 24.]

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," May 2015. [Pages ix, 24, 25, 31, 36, and 37.]

[42] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432. [Pages 24, 25, and 36.]

[43] Tensorflow. [Online]. Available: https://www.tensorflow.org/ [Page 25.]

[44] Pytorch. [Online]. Available: https://pytorch.org/ [Page 25.]

[45] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "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. [Page 25.]

[46] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021. [Page 25.]

[47] N. Siddique, P. Sidike, C. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: theory and applications," *arXiv preprint arXiv:2011.01118*, 2020. [Page 25.]

[48] B. McCrindle, K. Zukotynski, T. E. Doyle, and M. D. Noseworthy, "A radiology-focused review of predictive uncertainty for ai interpretability in computer-assisted segmentation," *Radiology: Artificial Intelligence*, vol. 3, no. 6, p. e210031, 2021. [Pages 25, 26, and 27.]

[49] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017. [Pages 26 and 37.]

[50] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015. [Pages 26 and 27.]

[51] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017. [Page 26.]

[52] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. Eslami, D. Jimenez Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," *Advances in neural information processing systems*, vol. 31, 2018. [Page 26.]

[53] M. Y. Avci, Z. Li, Q. Fan, S. Huang, B. Bilgic, and Q. Tian, "Quantifying the uncertainty of neural networks using monte carlo dropout for deep learning based quantitative mri," *arXiv preprint arXiv:2112.01587*, 2021. [Pages ix, 27, and 28.]

[54] M.-H. Laves, S. Ihler, K.-P. Kortmann, and T. Ortmaier, "Well-calibrated model uncertainty with temperature scaling for dropout variational inference," *arXiv preprint arXiv:1909.13550*, 2019. [Pages 30, 39, 40, 60, and 61.]

[55] Nsclc-radiomics. [Online]. Available: https://wiki.cancerimagingarchive.net/display/Public/NSCLC-Radiomics#16056854cf0f73682c354d8884fa7128434b6217 [Page 32.]

[56] H. J. W. L. Aerts, L. Wee, E. Rios Velazquez, R. T. H. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, F. Hoebers, M. M. Rietbergen, C. R. Leemans, A. Dekker, J. Quackenbush, R. J. Gillies, and P. Lambin, "Data from NSCLC-Radiomics," in *The Cancer Imaging Archive*, 2019. [Page 32.]

[57] Torchio. [Online]. Available: https://torchio.readthedocs.io/ [Page 35.]

[58] H. Lu, H. Wang, Q. Zhang, S. W. Yoon, and D. Won, "A 3d convolutional neural network for volumetric image semantic segmentation," *Procedia Manufacturing*, vol. 39, pp. 422–428, 2019. [Page 36.]

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014. [Page 37.]

[60] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT express*, vol. 6, no. 4, pp. 312–315, 2020. [Page 37.]

[61] Image slices viewer. [Online]. Available: https://matplotlib.org/3.1.1/gallery/event_handling/image_slices_viewer.html [Page 39.]

[62] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, no. 1-2, pp. 12–22, 1983. [Page 39.]

[63] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330. [Page 39.]

[64] dicompyler. [Online]. Available: https://www.dicompyler.com/ [Page 40.]

[65] Otsu's method. [Online]. Available: https://en.wikipedia.org/wiki/Otsu%27s_method [Page 41.]

[66] K. Wickstrøm, M. Kampffmeyer, and R. Jenssen, "Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps," *Medical image analysis*, vol. 60, p. 101619, 2020. [Page 59.]

[67] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li, "Mitigating neural network overconfidence with logit normalization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 631–23 644. [Page 60.]

[68] What to do when your ml model suffers from overconfidence? [Online]. Available: https://cio.economictimes.indiatimes.com/news/next-gen-technologies/what-to-do-when-your-ml-model-suffers-from-overconfidence/87500302?redirect=1 [Page 60.]

[69] Y. Li and M. Zhang, "Tier-a: Denoising learning framework for information extraction," *arXiv preprint arXiv:2211.11527*, 2022. [Page 60.]

[70] Artificial intelligence: Overfitting. [Online]. Available: https://artint.info/2e/html/ArtInt2e.Ch7.S4.html [Page 60.]

[71] Z. Ding, X. Han, P. Liu, and M. Niethammer, "Local temperature scaling for probability calibration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6889–6899. [Page 61.]

Appendix A

Clinical Evaluation Results

Table A.1: Results from clinical experiment.

Patient number	MU	RUV	Clinical evaluation
0	0,0740	0,4810	H
1	0,0598	0,2708	L
2	0,0914	0,6165	I
3	0,0507	0,2760	L
4	0,0504	0,2849	L
6	0,0481	0,2954	L
7	0,0538	0,2598	L
8	0,0921	0,5195	H
9	0,0632	0,5970	L
10	0,0524	0,3893	L
11	0,1122	1,9244	H
12	0,0744	0,3320	H
13	0,1254	1,7947	H
14	0,0617	0,2523	I
15	0,0629	1,5098	I
16	0,0511	0,4194	I
17	0,0904	0,5879	H
18	0,0730	0,4446	I
19	0,0746	0,6154	L
20	0,0736	1,1421	L
21	0,0854	0,5462	I
22	0,0890	0,4719	H
23	0,0817	0,7444	H
24	0,0531	0,3866	L
25	0,0587	0,4408	L
26	0,0770	0,5281	H
27	0,0777	0,5060	I
28	0,0671	0,3324	L
29	0,0995	0,7907	I
30	0,0467	0,2703	I
31	0,0733	0,4977	H
32	0,0628	0,8927	H
33	0,0563	0,3941	L
34	0,1001	1,7168	H
35	0,0815	0,6416	I
36	0,0640	0,2539	I
37	0,0437	0,2649	L
38	0,1140	2,4451	H
39	0,0507	0,1987	I
40	0,0702	0,3708	I
41	0,0892	0,9386	H

Appendix B

Otsu thresholds

Table B.1: Threshold to extract the high uncertainty area with Otsu thresholding approach.

Testset patient number	Otsu Threshold
0	0,285
1	0,277
10	0,287
11	0,283
12	0,286
13	0,288
14	0,277
15	0,284
16	0,277
17	0,282
18	0,275
19	0,283
2	0,285
20	0,267
21	0,285
22	0,280
23	0,284
24	0,291
25	0,280
26	0,287
27	0,279
28	0,282
29	0,284
3	0,286
30	0,261
31	0,286
32	0,283
33	0,290
34	0,284
35	0,283
36	0,274
37	0,260
38	0,291
39	0,280
4	0,285
40	0,285
41	0,287
6	0,269
7	0,286
8	0,287
9	0,283