UCLouvain

École polytechnique de Louvain

Deep learning to predict optimized dose and its uncertainty for radiation oncology treatment planning

Author: **Alyssa Vanginderdeuren**
Supervisors: **John Lee, Ana Maria Barragan Montero**
Readers: **Christophe De Vleeschouwer, Margerie Huet**
Academic year 2020–2021
Master [120] in Mathematical Engineering

Abstract

In recent years, deep learning has seen its potential significantly increased in the medical field, with various applications making the work of physicians more automated and efficient. One area of application where neural networks have a crucial role to play is in radiotherapy, one of the most common cancer treatments.

Every radiotherapy treatment is preceded by a treatment planning phase in which an important step is to optimise the dose of radiation received by the patient. This time-consuming task could be automated using a convolutional neural network based on the patient's anatomical parameters. In this master thesis, we will use the well-known U-Net architecture for this task.

However, this model would be used as a black box predicting the dose distribution with no information on the certainty of the model on its prediction. In this work, we propose to accompany these models with a map indicating, at each voxel, the level of uncertainty of the model. To do this, a Bayesian model is too costly. This is why we developed three simple methods that can be easily adapted to other architectures: the Monte-Carlo dropout, the bootstrap method and a modification of the U-Net. These methods were tested on a database of 200 patients suffering from head and neck cancer.

A correlation between our measure of uncertainty and the error made by the model of about 0.27 was found on the planning target volume (PTV) and 0.56 outside. A theoretical correlation of 0.71 was computed outside the PTV.

Acknowledgements

First of all, I would like to thank the members of the MIRO lab for integrating me into the team with kindness during this project.

I would like to sincerely thank my promoter, John Lee, for proposing this project, trusting me, and always giving me good advice.

A huge thanks to Ana and Margerie for their support and advice. I could always turn to them to find the answer to my questions.

Finally, I would like to thank my parents, friends and boyfriend for their encouragement throughout this year.

Contents

Acronyms

ADD actual dose distribution.

BNN Bayesian Neural Network.

CNN Convolutional Neural network.

CT Computed Tomography.

CTV Clinical Target Volume.

DVH Dose-volume histogram.

GPU graphics process unit.

GTV Gross Target Volume.

Gy Gray.

MAE mean absolute error.

MCD Monte-Carlo Dropout.

MSE mean squared error.

NN Neural network.

OAR organ at risk.

PDD predicted dose distribution.

PTV Planning Target Volume.

ROI region of interest.

SD standard deviation.

VOI volume of interest.

Chapter 1

Introduction

In 2020, 83 267 new cancer cases were diagnosed in Belgium, 19 292 789 worldwide [1]. There already exist various treatments for cancer, such as surgery, radiotherapy, chemotherapy, hormone therapy, immunotherapy, and other new treatments that can also be combined for better efficiency. Among them, radiotherapy, curative or palliative, is involved in more than half of cancer treatments. In most cases, external radiotherapy is preferred to brachytherapy, where radioactive sources are implanted directly inside the patient's body. External radiotherapy uses photons (e.g., X-rays) in 90% of cases but can also use particles like electrons or protons, irradiating the tumour with ionising radiation, damaging the DNA of the cancer cells until they die. As cancer cells have a reduced capacity for regeneration, treatment sessions are spread over several days. Between sessions, healthy cells can regenerate more than the cancer ones and thus better resist the treatment.

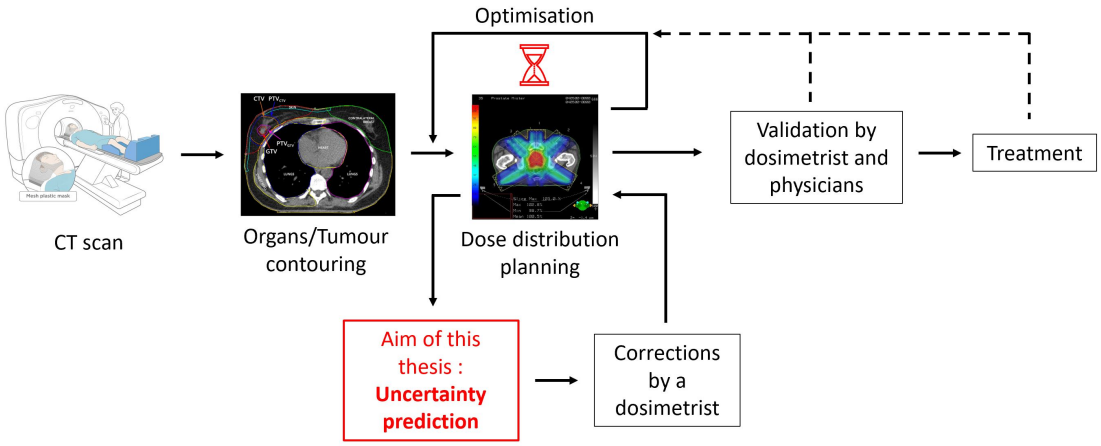


Figure 1.1: Treatment planning steps.[2][3]

In recent years, deep learning has seen its potential significantly increased in the medical field with various applications making doctors' work more automated and saving precious time. Radiotherapy is one of the areas where neural networks have already shown good results.

Every radiotherapy treatment is preceded by a treatment planning phase that requires the intervention of several specialists. As shown in Figure 1.1, the first step is to make a CT scan of the patient on which the contours of the organs at risk and of the tumour are drawn.
On this basis, the dosimetrist can optimise the dose distribution. This is a time-consuming step requiring optimisation software and several iterations before the final result is approved by the clinical team. Some researchers proposed to automate this task using convolutional neural networks based on the patient's anatomical data. A well-known architecture for this process is the U-Net [4]. A model is trained using a sample of patients for whom the optimal dose was previously calculated manually.

However, introducing such a model into a planning process would lead the specialist to use it as a black box predicting the dose distribution without information about the model's confidence in its prediction. In this work, we propose to accompany these models with a map indicating, at each voxel, the level of uncertainty of the model. Such a map would improve the interpretability of the prediction and would make it possible to correct areas of high uncertainty when they are critical, for example, on an organ at risk or on planning target volume, to improve the prediction. To do this, a Bayesian deep network being too costly (see Section 2.2.3), we have developed three simple methods that can be easily adapted to other architectures: the Monte-Carlo dropout, the bootstrap method, and an adaptation of the U-Net detailed in Chapter 4. These methods were tested on a database of 200 head and neck cancer patients.

The rest of this document is organized as follows. Section 2 of this paper sets the context and explains the treatment planning phase in more detail. It also provides an introduction to deep learning and its application in treatment planning. Then, Section 3 reviews previous work that has been published in the literature. Section 4 describes the methods proposed in this thesis, followed by the results in Section 5. Finally, a discussion and a conclusion close the document in Sections 6 and 7.

Chapter 2

Applicative Context: Radiation Oncology and Deep Learning

Radiotherapy can be used as a curative procedure to treat a cancer entirely or in combination with other treatments to make them more efficient, for example with chemotherapy or before surgery. It can also be prescribed after surgery to reduce the risk of cancer recurrence. Finally, radiotherapy can be palliative, relieving symptoms when a cure is not possible.

External radiation therapy involves irradiating the tumour through the patient's body with ionising radiation (photons, electrons, protons). When the rays come into contact with the body's cells, the energy is transferred to them and causes damage to the DNA or even causes the death of the cell if the radiation dose is high enough. The dose of radiation delivered is measured in Gray (Gy), one Gray representing the energy of ionising radiation delivering an energy of one joule to a homogeneous medium with a mass of one kg [5]. Although they are not directly targeted, the organs surrounding the tumour are also irradiated at lower dose that may induce undesirable side effects even after the treatment.

2.1 Treatment planning

Before a patient can receive any treatment, a patient-specific plan must be established. This process involves different steps that require the collaboration of physicians, radiologists and a dosimetrist.

2.1.1 Computed tomography scan

The first step is to have the patient undergo a Computed Tomography (CT) scan to obtain an image of the area to be treated, its organs and their position in relation to the tumour. A CT scan is a 3D image of the patient formed by assembling many 2D images, each representing a layer of the patient (Figure 2.1). During a CT scan, the patient lies on a table that moves slowly through the scanner. An X-ray tube rotates around the patient and sends a narrow beam of X-ray through the body. X-rays are captured by a digital detector on the opposite side of the source. After each complete rotation of the tube, a computer reconstructs a 2D image. Depending on the type of tissue they pass through, X-rays will be more or less attenuated. The most attenuated tissues appear the lightest (bones appear white). The system is presented in Figure 2.2

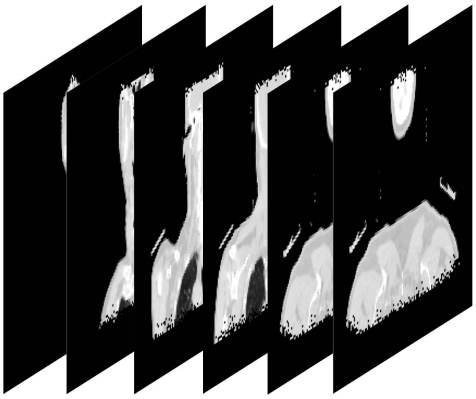


Figure 2.1: 3D CT scan obtained by assembling 2D images for each slice of the patient.

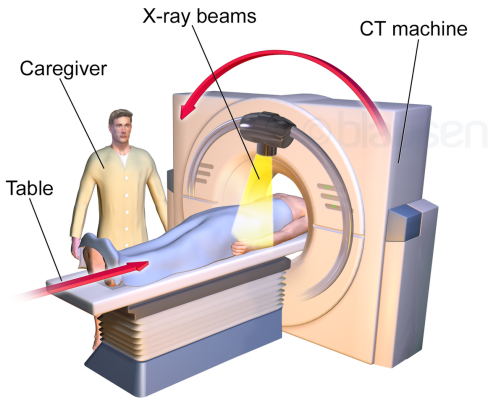


Figure 2.2: CT scan process. The table on which the patient is lying passes slowly through the ring. The X-ray source rotates around the patient. X-rays are captured by a digital detector and processed by a computer to produce 2D images.[6]

It is essential that the patient keeps exactly the same position during the scan and the treatment, as the radiation parameters will be determined on the basis of the scan in order to optimise the dose delivered. A mismatch with the patient's anatomy during the treatment could deteriorate the expected results. To achieve this, different techniques can be used to maintain the patient's position depending on the area to be scanned. One of these is a plastic mask molded on the patient's

head and attached to the table to hold it in place (see Figure 2.3). The mould will be made before the scan and reused throughout the treatment.

Figure 2.3: Mask used during the scan to keep the patient's head in the same position during the scan and treatment.[7]

2.1.2 Organs and tumour delineation

The second step is to delineate the different organ at risk (OAR) and the tumour (Figure 2.4). The tumour area to be treated is divided into several sub-areas as shown in Figure 2.5 which are irradiated with an appropriate intensity.

Gross Target Volume (GTV) : delineates the primary tumour that is palpable or visible on medical imaging.

Clinical Target Volume (CTV) : covers the GTV and microscopic extensions of the tumour that are not visible on imaging.

Planning Target Volume (PTV) : corresponds to the CTV to which a safety margin is added to account for positioning uncertainties, possible organ and patient movements, and dose homogeneity problems within the CTV.

The dose prescription may vary in different areas of the PTV. In this thesis, up to three irradiation levels can be encountered: 56, 63, and 70 Gy.

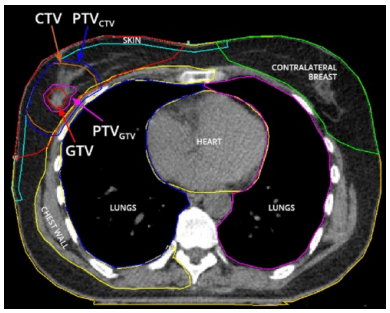


Figure 2.4: Example of a CT scan on which the contours of the organs and the tumour have been drawn.[8]

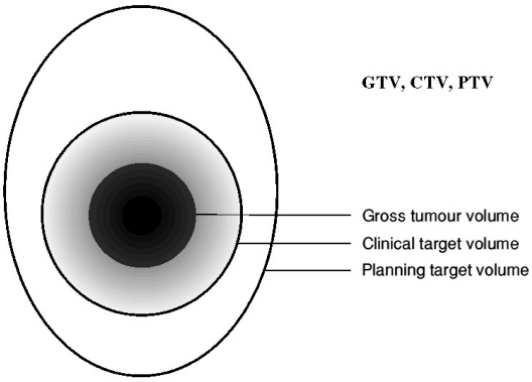


Figure 2.5: Different common tumour delineations : GTV, CTV and PTV on the CT scan of the patient.[9]

2.1.3 Dose planning optimisation

Once these steps have been completed, a dosimetrist is responsible for determining the optimal radiation parameters to achieve the best possible dose distribution. This is an iterative process using optimisation software. An example of such software is RayStation, commercialised by RaySearch Laboratories [10]. At each step, the program receives criteria in the form of a manually entered objective function. This function is a weighted sum of several criteria on the dose distribution that we would like to respect as much as possible, especially on the OAR and PTV. Dose limit recommendations for the different OAR to minimise side effects are available in the literature. An example is shown in Figure 2.6. The optimisation of this objective function based on the patient's anatomy allows us to obtain the beam parameters to be used automatically. Once the optimal solution has been found, the dose distribution is evaluated by a dosimetrist (and physicians). If it is considered good enough, it is kept and approved. If not, an additional iteration by adjusting the weights of the objective function is necessary. This method is called inverse planning because the beam parameters are computed based on constraints and the patient's anatomy. In contrast, forward planning determines the dose distribution based on manually set beam parameters but is no longer relevant. This iterative process can be very long (several minutes/hours). Therefore, using a deep neural network is a major asset in treatment planning as it allows to determine the optimal dose distribution to be achieved in seconds. The iterative process (tuning the function to be optimised) in inverse planning is replaced by optimising the beam parameters to get as close as possible to the predicted dose distribution, which is cheaper (see Dose mimicking in Section 2.3.2).

With the plan in place, the treatment can start. It is possible that during the treatment process, reality deviates too much from the predictions and the plan needs to be revised. Here again, the automation of the computation is a considerable advantage as it is cheap and allows the predictions to be recalculated more frequently in order to better match reality throughout the treatment.

Organ	Endpoint	Rate (%)	Dose-volume parameter	D_{max} (Gy)	D_{mean} (Gy)
Brain	Symptomatic necrosis	<3 <5		<60 <65	
Brainstem	Necrosis or cranial neuropathy	<5 <5	D100 <54 Gy D1–10 cc ≤59 Gy	<64 Point	
Spinal cord	Grade ≥2 myelopathy	<1		50	
Optic nerve & chiasm	Optic neuropathy	<3 3–7		<55 55–60	<50
Retina	Blindness	<1		<50	
Cochlea	Hearing loss	<15			≤45
Parotid 1	Grade 4 xerostomia	<20			<20
Parotid 2		<20			<25
Mandible	ORN	<5		<70 Point	
Pharyngeal constrictors	PEG tube dependent Aspiration	<5 <5			<50 <60
Larynx	Grade ≥2 edema	<20	V50 <27%		<44

Figure 2.6: Recommended tolerance of different organs to radiation in order to avoid side effects due to excessive dose.[11]

2.1.4 Prediction Evaluation

Several metrics can be used to compare the prediction to the actual dose delivery.

D_x is the maximum dose delivered to x% of the volume of interest (VOI). Common metrics are D_2, D_5, D_{95}, D_{98}, D_{mean} (average dose on VOI) and D_{max} (maximum dose on VOI). Metrics such as D_{98}, D_{mean} are used on the PTV because it should receive a sufficient dose of radiation. D_2 and D_{max} are used on organs at risk to check that they do not receive an excessive dose. V_x can also be measured and indicates the volume receiving a dose of x Gy.

Based on this, we can construct the Dose-volume histogram (DVH) (Figure 2.7). For each OAR and each dose prescription of the PTV, the curve shows the percentage of the VOI receiving the dose on the x-axis (solid line for the ground truth and dashed line for the prediction).

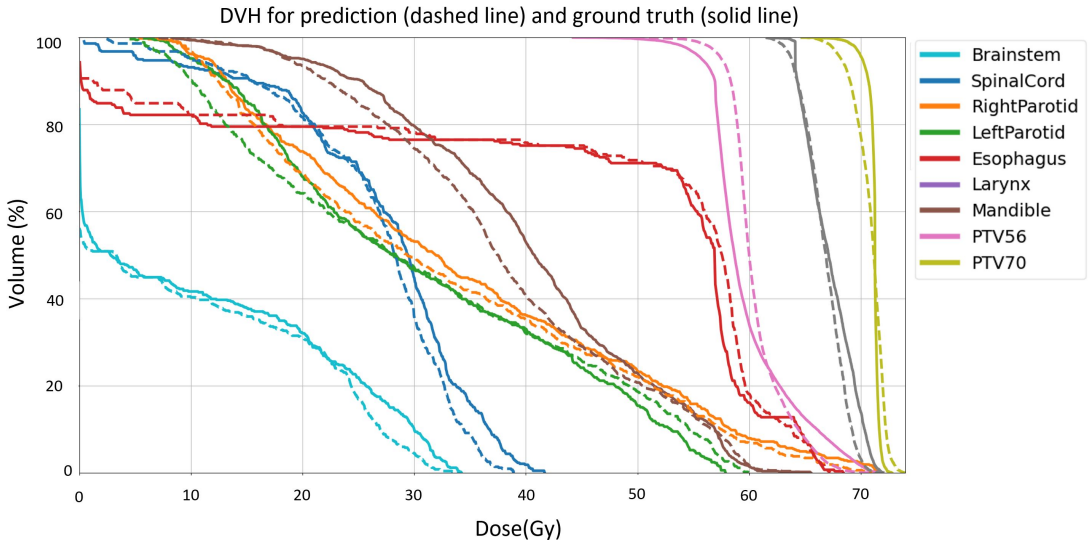


Figure 2.7: Dose-volume histogram : for each dose on the x-axis, the curve shows the percentage of the volume of interest receiving that dose.

2.2 Deep Learning

This section will first review the evolution of deep learning by giving a short historical introduction. Then, we will present the basics of deep learning and Bayesian neural networks and develop the key elements that constitute a convolutional neural network, also used in this work.

"Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network."[12]

2.2.1 History

The origins of deep learning go back to 1943 with the first description of the artificial neuron by Warren Sturgis McCulloch and Walter Pitts, inspired by the structure of a real neuron (Figure 2.8), marking the beginning of a golden period for the development of artificial intelligence. In 1957, Frank Rosenblatt introduced the Perceptron, the first implementation of the McCulloch and Pitts neuron, followed,

in 1965, by Alexey Grigorevich Ivakhnenko, who proposed the first deep network. Unfortunately, due to criticism about the limitations of single-layer perceptrons in 1969 (e.g. perceptrons are unable to learn and approximate the XOR logic function), the popularity of AI declined. The appearance of gradient backpropagation by P.J. Werbos in 1974 reignited the interest in AI. Discoveries continued with the proposal of the Neocognitron by Kunihiko Fukushima in 1980, followed by the convolutional neural network (Yann Le Cun, 1989), auto association in artificial neural networks trained with backpropagation (Kramer, Oja, DeMers, Cottrell, 1991-1993), transfer learning (Pratt, 1991-1993) and max-pooling (Weng J., Ahuja N., Huang T.S.) in 1993. Then, the vanishing gradient problem and the lack of sufficient computing power affected the further development of AI. The arrival of GPU (Graphical Processing Unit) generated a renewed interest in 2004, giving rise to new architectures such as AlexNet (Alex Krizhevsky, 2012), ResNet (He K., Zhang X., Ren S., Sun, J) and U-Net (Olaf Ronneberger) in 2015 or DenseNet (Huang G., Liu Z., van der Maaten L. and Weinberger K.Q) in 2016, with growing interest until today. The eventful history of deep learning is presented as a timeline in Figure 2.9. [13]

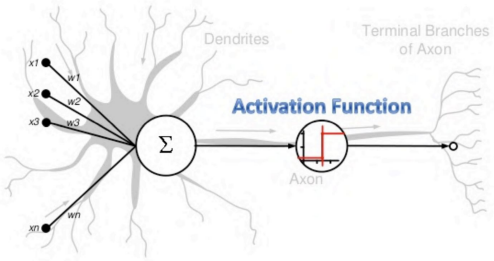


Figure 2.8: Representation of an artificial neuron introduced by Warren Sturgis McCulloch and Walter Pitts (1943).[13]

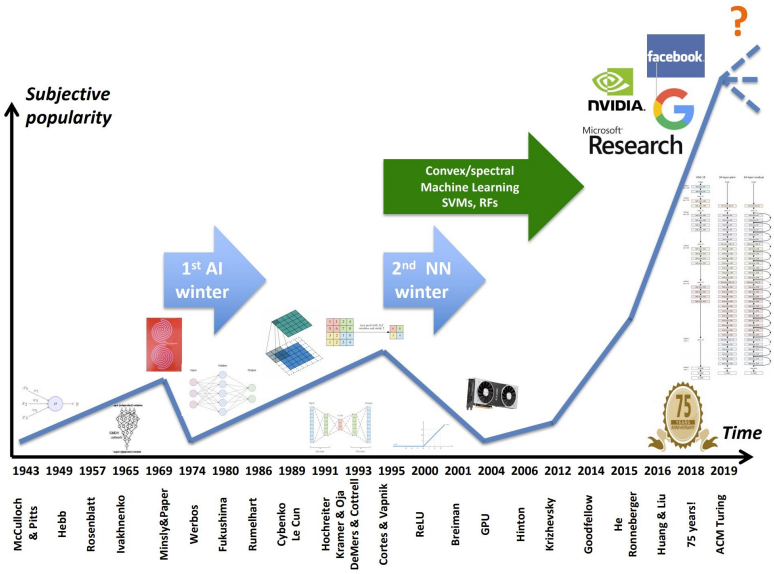


Figure 2.9: Timeline representing the interest in AI according to different important discoveries.[13]

2.2.2 Deep Neural Networks

A deep neural network is a network composed of several layers, each composed of neurons, the basic element of machine learning networks (Figure 2.10). A neuron is a computational unit, it receives several input values from the previous layer and returns an output value (Figure 2.11). The first layer is called the *Input layer*, the last layer is the *Output layer*, and the intermediate layers are the *Hidden layers*. The neurons of the successive layers are connected by edges to which a weight is associated. During training, these weights are optimised to minimise the final error.

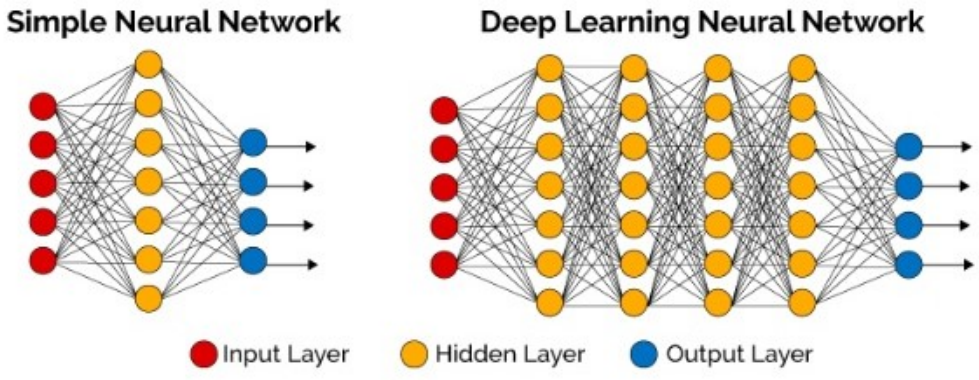


Figure 2.10: Representation of Neural Network and Deep Neural Network layers.[14]

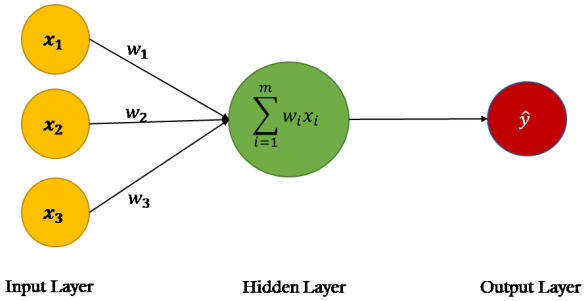


Figure 2.11: Schematic representation of the functioning of a neuron in machine learning.[15]

2.2.3 Bayesian Neural Networks

Bayesian Neural Network (BNN) are a combination of deep neural networks and probabilistic models. Unlike a classical neural network where the weights (parameters) have fixed values, the weights of BNNs are characterised by a probability density function. The difference between a neural network and a BNN is illustrated in Figure 2.13.

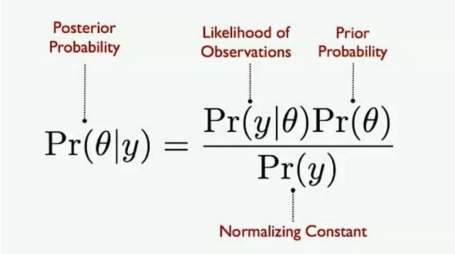


Figure 2.12: Bayes' Theorem.[16]

The weights of the network are updated according to Bayes' theorem (Figure 2.12). In this formula, the weights are the parameters of our problem, represented by the variable θ. y is the output of the BNN, result of the model prediction for an input x. The posterior probability can be updated as the training data is presented to the model.

The first difficulty with this type of neural network is providing the model with a prior probability function before starting to train it. Secondly, the main drawback of BNNs is the computational cost required to train them due to a large number of parameters.

The advantage of BNN is that it allows to quickly obtain a measure of the uncertainty of the model thanks to its probability distribution on the output.

Reference [16] provides a more detailed introduction to BNNs.

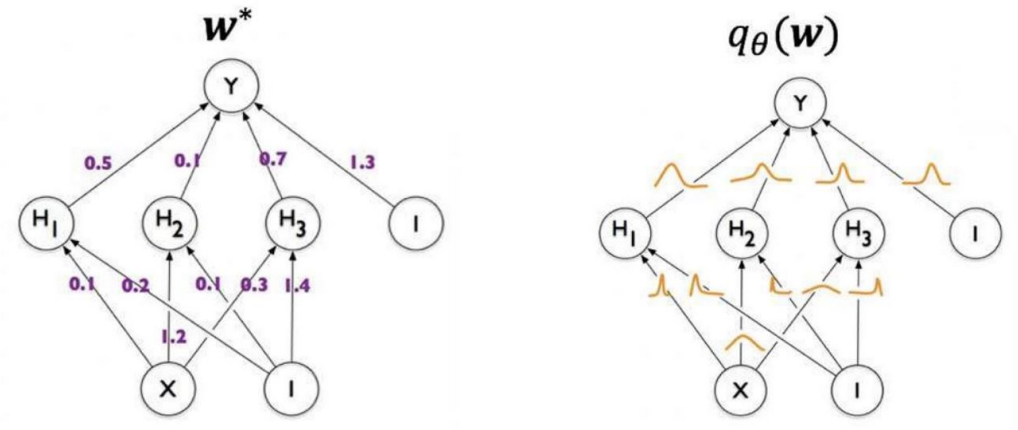


Figure 2.13: Representation of a classical Neural Network on the left and a Bayesian Neural Network on the right. In a NN, weights are fixed values while in a BNN, weights are characterised by a probability density function.[16]

2.2.4 Convolutional Neural Networks

Convolutional Neural network (CNN) adapt the deep neural networks presented in section 2.2.2 to take images as input. Indeed, using a neural network with an image would require too many parameters (weights) to be optimised. The CNNs are composed of layers and perform various operations to extract helpful information from the input images. The size of the images is also reduced throughout the layers to reduce the number of parameters in the network. The main layers applied (named according to the operation they perform) are: convolution layer, pooling layer, ReLU correction layer, fully-connected layer. A representation of a CNN for a classification task is shown in Figure 2.14.

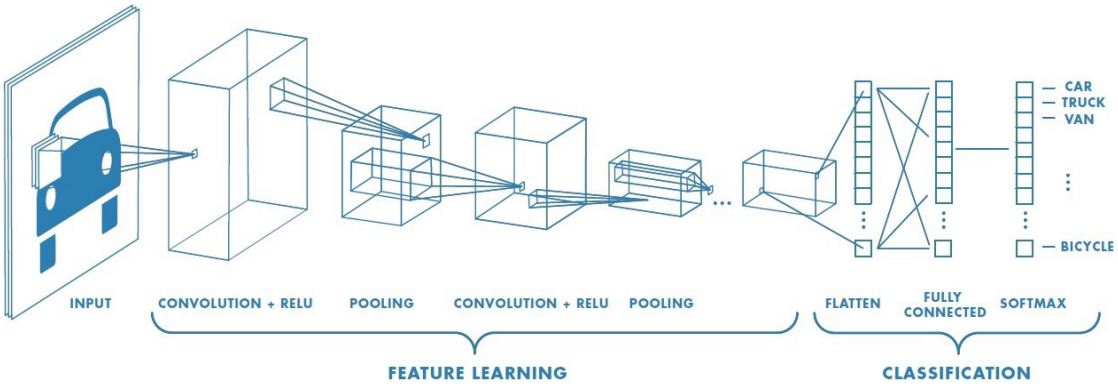


Figure 2.14: Convolutional Neural Network architecture for image classification.[17]

2.2.5 Convolution

$$C(x,y) = \sum_{dx=-s}^{s} \sum_{dy=-s}^{s} M(x+dx, y+dy)w(dx,dy) \tag{2.1}$$

The convolution operation is used to extract some important information from the images. It consists of moving a filter through the image from the top left corner to the bottom right corner according to the Formula 2.1 where C is the resultant matrix, M the input matrix, w the filter used of size nxn (n odd) and $s = \frac{n-1}{2}$. In the U-Net architecture (Section 3.1), the first convolution layers are used to extract low-level features such as edges, circles, gradient orientations,... while the deeper layers are used to detect high-level features, i.e. more complicated patterns (e.g. the nose, the eyes for a segmentation task).

Figure 2.15: Original image on the left and result of applying an edge detector filter to this image on the right.[18]

The weights associated with a filter determine the feature that it will be able to detect. For example, Figure 2.15 shows the effect of applying a filter adapted to edge detection. The adaptation to the 3D case is made simply by adding a dimension to the filter, each feature map being treated independently.
The opposite operation also exists and is called deconvolution.

2.2.6 Max Pooling

Max pooling is a downsampling operation in which a filter is applied to an initial matrix. As shown in Figure 2.16, for 2x2 max pooling as in the U-Net, the filter is moved with a stride of 2 and the largest value is retained for each filter position. In 3D, max pooling is applied to each of the features, so their number remains unchanged (Figure 2.17). This reduces the resolution of the images and, consequently, the number of parameters to be optimised. It decreases the required computing power and ensures a faster convergence in the learning phase. It is also possible to do the opposite operation, called unpooling.

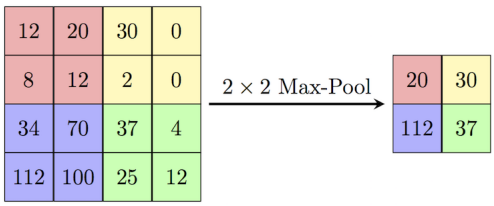


Figure 2.16: Example of max pooling in 2D.[19]

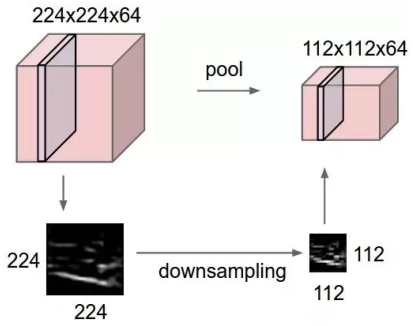


Figure 2.17: Max pooling operation in 3D.[19]

2.2.7 Activation function

Neural networks that do not contain layer activation are linear. The activation function makes them non-linear, which allows to learn more complex tasks. In CNNs, the activation functions usually follow the convolution layers. The ReLU function is a piecewise linear function typical of neural networks. It returns the input if it is positive and zero otherwise. Its advantage is that it is easy to evaluate and avoid the gradient vanishing problem, allowing faster network training. Other functions are frequently used, such as the sigmoid function or the tanh function. However, they have several disadvantages, such as saturation (constant beyond a range of values centred at 0) and their sensitivity only in the centre of their domain. Another essential drawback is that they do not prevent the gradient vanishing problem. Figure 2.18 shows the three discussed functions: ReLU, sigmoid and tanh.

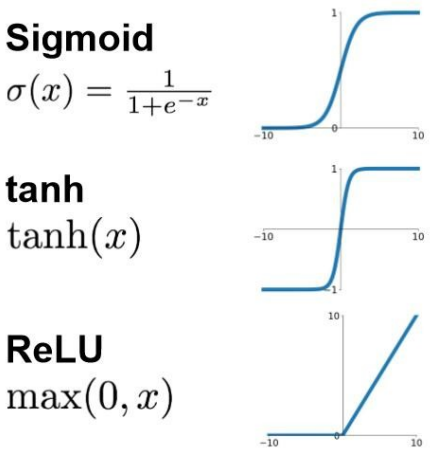


Figure 2.18: Activation functions : Sigmoid, Tanh and ReLU.[20]

2.2.8 Dropout regularization

When training a deep neural network, the amount of data may be too small compared to the number of network parameters. An example is provided in Figure 2.19. In this case, the model becomes too specialised for the particular case of the data on which it has been trained and no longer gives correct results for a new set of data. In order to make this model more general, it is possible to use dropout during training. When dropout is active, a dropout rate p is applied, i.e. each neuron can be switched off with probability p. When a neuron is switched off, its output is zero and its weight has no more impact on the model (Figure 2.20).

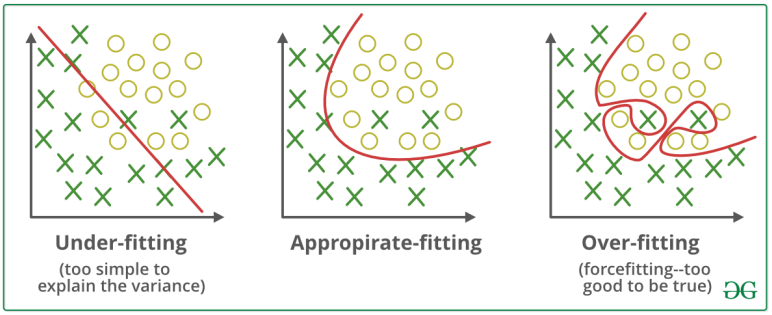
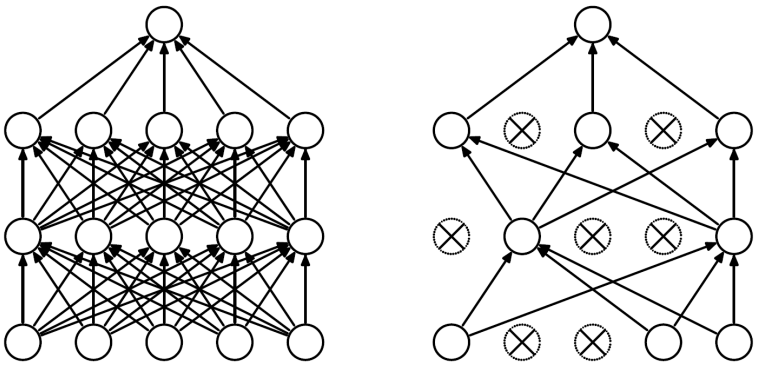


Figure 2.19: Illustration of overfitting.[21]



(a) Standard Neural Net (b) After applying dropout.

Figure 2.20: Dropout effet on a neural network.[22]

2.3 Automated treatment planning

2.3.1 Automated steps

Many studies have looked at automating the different treatment planning steps to improve its efficiency and planning time. As can be seen in Figure 2.21, the delineation phase of OAR and PTV can be done with a deep neural network trained for the segmentation task.

In this work, we focused on the dose distribution planning phase. Again, deep learning is used to predict the optimal dose distribution based on the anatomical data of a new patient using a previously trained model. This step takes only a few seconds/minutes and avoids the long iterative process previously presented.

The major contribution of this thesis is the loop containing the measurement of the model uncertainty. This measure can be calculated in several ways detailed in Chapter 4. Once calculated, it can be passed to the dosimetrist, who can manually intervene to correct specific areas. It allows for a better understanding and interpretability of the results and reinforces the reliability of the results, which is very important in such treatments.

The human input of this treatment planning is reduced thereby freeing up more time for the specialists to carry out more complex and less general tasks that require their expertise.

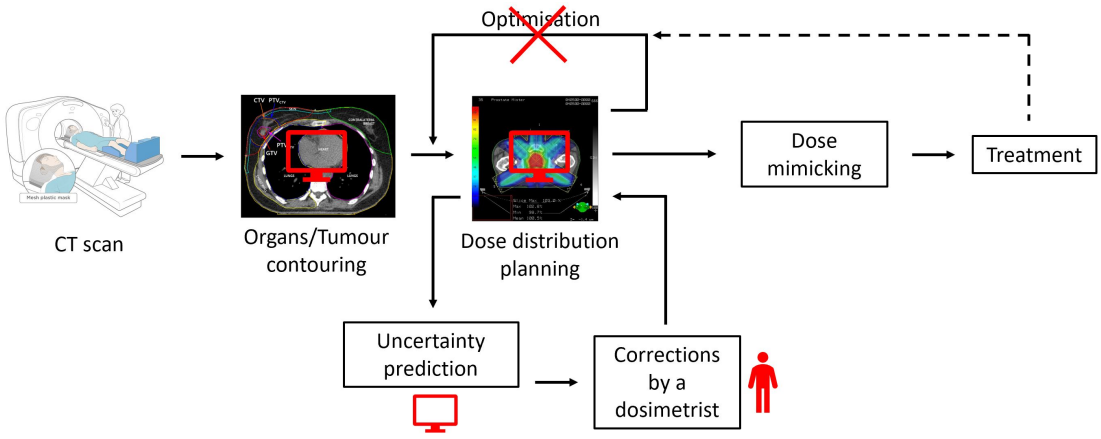


Figure 2.21: Automated treatment planning steps.

2.3.2 Dose mimicking

The automation of the planning process with the help of deep learning allows us to obtain an optimal distribution dose for each new patient. The model only returns the dose distribution but not the beam parameters to obtain it. Therefore, an additional step is necessary to determine them and obtain a distribution dose as close as possible to the predicted optimal one. [23]

RaySearch Laboratories have developed an algorithm that uses machine learning to automate the treatment planning process [24]. As shown in Figure 2.22, The process is different from the traditional process with inverse planning explained earlier because the dose distribution calculation no longer requires iterations with human intervention (time-consuming). In the machine learning workflow, the dose distribution is predicted by a trained neural network model. Then, the beam parameters are optimised to reproduce the predicted dose distribution as well as possible. This step leads to the final dose distribution, which closes the treatment planning phase.

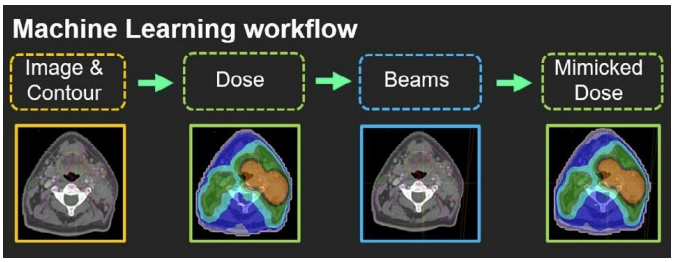


Figure 2.22: Machine learning process from RaySearch Laboratories. Based on the annotated CT scan of the patient, a model predicts the optimal dose distribution. Then, by dose mimicking, the beam parameters are optimised to reproduce the predicted dose distribution as closely as possible. The distribution dose with the determined parameters is calculated, and the treatment can start.[24]

Chapter 3

Literature Review

In recent years, following the growing interest in deep learning, various architectures have been proposed to predict the dose distribution during treatment planning. One of them, U-Net, is presented in Section 3.1. Secondly, an overview of uncertainty calculation methods existing in the literature is proposed (Section 3.2.1). We conclude with an introduction to active learning (Section 3.2.2).

3.1 Dose distribution with U-Net

U-Net is a convolutional neural network architecture composed of 5 layers. It consists of 3 parts as shown in Figure 3.1: the contracting path or encoding part on the left, the bottleneck and the expansion path or decoding part on the right, giving it its U shape from which its name is derived. The contracting path extracts information from the input images and reduces their resolution (downsampling path) for more efficient processing. It consists of the repetition of two 3x3 convolutions followed by a ReLU activation function and a 2x2 max pooling operation to move to the lower layer. In the centre of the U, the bottleneck is composed of two 3x3 convolutions and a 2x2 upsampling. Finally, the decoding part allows to retrieve a complete image at the output thanks to the upsampling. The decoding part follows the same scheme as the encoding one with two convolutions, but the downsampling is replaced by a 2x2 upsampling operation. We also notice that, at each level, the input of the expansive path is appended to the corresponding output of the contracting part before the convolutions to recover the spatial information lost during downsampling. Skip connections can also play a role in gradient vanishing by creating shortcuts [25]. More information on the different operations can be found in Chapter 2.

The U-Net architecture was introduced in 2015 by Olaf Ronneberger et al. in [4] for

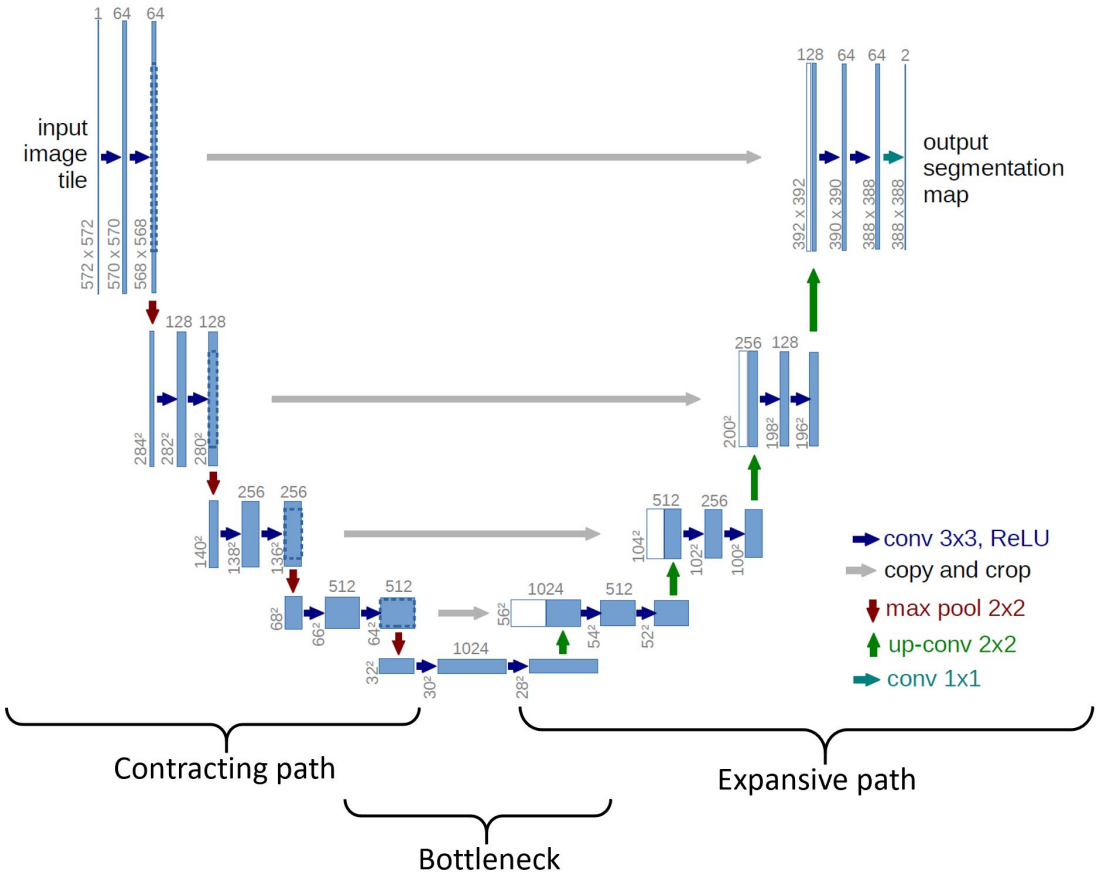


Figure 3.1: First U-Net architecture by Ronneberger et al. for segmentation.[4]

image segmentation. It is an adaptation and improvement of the fully convolutional network [26]. With this architecture, they won the ISBI cell tracking challenge in 2015. Its advantage over previous CNNs is the reduced number of images needed to train the model, thanks to a strong use of data augmentation. Segmentation is also faster and more accurate. Many feature channels in the upsampling part allow the context information to be propagated to higher resolution layers. The use of elastic deformations for data augmentation teaches the network invariance to such deformations.

This architecture was then reused and adapted by Nguyen et al. in 2017 in order to predict the optimal dose distribution in radiotherapy for prostate cancer patients [27]. As input, the patient's anatomical data are taken into account in the form of 6 contours, including the PTV (dose prescription) and various organs surrounding the tumour. The model is trained on single slices of the patient and each of the six inputs is processed independently. As the central slices containing the PTV

are the most important, they will be favoured by providing the slices to the model following a Gaussian distribution during training. Convolutional layers have been added (to have seven levels) and thus go from a 6-feature input of 256x256 pixels to features of size 4x4 at the bottom of the U-Net. Seven CNN layers were added at the output to arrive at a single filter. The authors were able to predict D_{max} and D_{mean} of all structures within 5.1% of the prescription dose.

In 2018, Nguyen et al. proposed a new architecture, Hierarchically Densely Connected U-net (HD U-Net) (Figure 3.2), based on the original U-Net and the DenseNet [28]. They are testing their architecture for patients with head and neck cancer (one of the most complicated cases) and are getting better results than either of the two individual architectures.

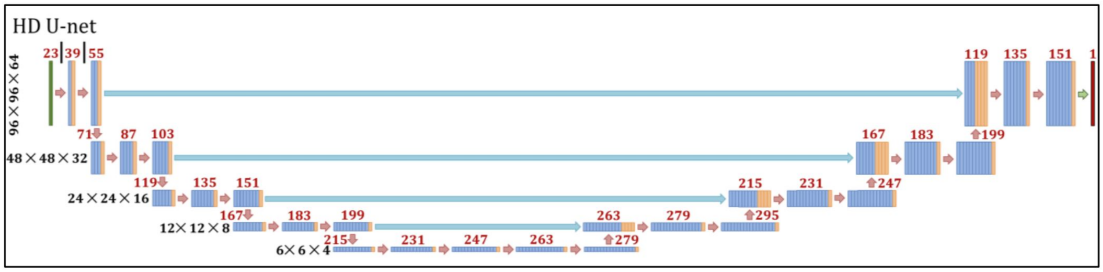


Figure 3.2: HD U-Net (Combination of U-Net with DenseNet by Nguyen et al. to predict dose distribution).[28]

In 2019, Barragán-Montero et al. generalised the HD U-Net model by taking into account variable beam configurations in [29]. The beam setup is thus added to the inputs. The model with beam configurations finally gives better results than taking only the patient's anatomy into account.

Other architectures were presented, sometimes as variants of the U-Net or in combination with it, for predicting the optimal dose distribution, such as the ResNet or DoseNet.[30][31]

3.2 Uncertainty in neural networks

Deep learning algorithms have undergone significant development in recent years, particularly in treatment planning for radiation therapy. In addition to the accuracy of these models, an important and still outstanding issue is their ability to express uncertainty to the professionals who use them.

We generally distinguish two types of uncertainty in machine learning [32]:

Aleatoric Uncertainty (irreducible) : refers to the notion of randomness, the variability of a response when the input is identical, the part of the uncertainty that could not be reduced by additional information. For example, if a prediction is made with a Neural network (NN) by activating the dropout during the prediction, there will always be an uncertainty in the prediction, which cannot be reduced.

Epistemic Uncertainty (reducible) : is due to a lack of knowledge. It concerns information that one does not know but that one could know. The epistemic uncertainty could be reduced with information input (e.g. by giving more training data to a NN, one could reduce the uncertainty of its prediction).

3.2.1 Monte Carlo Dropout, Bootstrap, Bayesian networks

Caldera and Nord compare in [33] three methods commonly used in deep learning to compute uncertainty, applied to a simple physical system: Bayesian neural networks (BNN), Monte-Carlo Dropout (MCD) called concrete dropout in the article and deep ensembles (DE), the equivalent of what we will call bootstrap in the following.

$$\hat{g} = \frac{1}{N}\sum_{i=1}^{N}\mu_i \tag{3.1}$$

$$\sigma_e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\mu_i^2 - \hat{g}^2} \tag{3.2}$$

For each method, predictions are made with different models. However, the technique used to obtain these models varies according to the method studied. The epistemic uncertainty is then calculated with Formulas 3.1 and 3.2 on the set of predictions, corresponding to the mean and standard deviation ($\hat{g}$ is the mean over the N predictions and σ_e the standard deviation). In this study, N=10 predictions are used.

BNNs are a class of neural networks in which each weight is characterised by a probability distribution rather than a fixed value (see Section 2.2.3). The uncertainty is assessed by taking samples from the posterior distribution. For the different parameter values, a prediction is made with the model. The uncertainty is finally calculated as the standard deviation on the different predictions. (Other articles have been published using BNNs to find uncertainty, such as [34][35].)

MCD is a method in which dropout is applied during training and testing. By applying the dropout during testing, each prediction will be made with a different model (the same model for which different neurons have been switched off). Each model gives a prediction based on which the uncertainty is calculated.

Finally, DE consists in training the same network several times with different training sets. Each training is a subset of the available training set whose elements have been randomly drawn with replacement. Each trained model will give a prediction. The uncertainty is calculated in the same way as for the other methods.

More recently, Nguyen et al. compared the bootstrap (equivalent of DE) and Monte-Carlo dropout methods for dose prediction in radiotherapy in [36] on the same principle as Caldera and Nord. 24 models were each trained on 160 of the 200 patients in the training set for the bootstrap. The architecture used is a U-Net.

$$\text{uncertainty}_{\text{raw}}(y, x, W_1, \ldots, W_T) \approx \sqrt{\frac{1}{T}\sum_{i=1}^{T} y(x, W_i)^2 - (\frac{1}{T}\sum_{i=1}^{T}(x, W_i))^2} \tag{3.3}$$

They propose a raw uncertainty corresponding to Caldeira's epistemic uncertainty (Equation 3.3) and a scaling factor for a region of interest (ROI). y(x,W_i) is the prediction made by a model whose weights are W_i (set of parameters) and input is x.

$$m_i = \frac{\text{dose}_{\text{pred},i} - \text{dose}_{\text{true},i}}{\text{uncertainty}_{\text{raw},i}} \tag{3.4}$$

$$m_{\sigma,ROI} = \sqrt{\frac{1}{n_{ROI}} \sum_{i \in ROI} m_i^2 - \left(\frac{1}{n_{ROI}} \sum_{i \in ROI} m_i\right)^2} \tag{3.5}$$

The scaling factor calculations are presented in Equations 3.4 and 3.5. It is calculated on the validation set and allows better interpretability of the uncertainty.

The $\text{uncertainty}_{\text{scaled}}$ is obtained by multiplying the scaling factor by the $\text{uncertainty}_{\text{raw}}$. Regarding prediction, the bagging method makes a lower error than the classical prediction. In terms of uncertainty, the scaled uncertainty of bagging has a better correlation with the error (mean absolute error on the prediction dose compared to the actual dose distribution) than the MCD, nevertheless remaining smaller or equal to 0.55.

3.2.2 Active learning

Active learning is a method of machine learning in which the algorithm can ask the user (oracle) to annotate the data of its choice. Annotating data to train a neural network is time-consuming, and the budget for it may be limited. Reference [37] proposed an optimised annotation method based on active learning. A baseline architecture is used to perform a task (e.g. classification, object detection, dose prediction). A loss prediction module is appended to this architecture to predict the main model's error during its task (Figure 3.3). The proposed algorithm is an iterative algorithm that works as follows. The set of labelled data points of size K at iteration i is denoted $\mathcal{L}_K^i$. The set of unlabeled data points of size K at iteration i is denoted $\mathcal{U}_K^i$. At the initial stage, we have an empty set $\mathcal{L}$ and a set $\mathcal{U}_N$, which contains the whole set of data points. K data points are randomly chosen in $\mathcal{U}$ and annotated. We now have $\mathcal{L}_K^0$ and $\mathcal{U}_{N-K}^0$ to begin the algorithm. At each iteration i (i$\geq$1), the main model and the loss module are trained using the set $\mathcal{L}^{i-1}$ as the training set. Then, the loss module is used to predict the error made at each data point. The algorithm selects the K unannotated points with the largest predicted error to query the oracle for annotation. These K annotated points leave the set $\mathcal{U}_{N-iK}^{i-1}$ that becomes $\mathcal{U}_{N-(i+1)K}^i$ to join the set $\mathcal{L}_{iK}^{i-1}$ that becomes $\mathcal{L}_{(i+1)K}^i$ for the next iteration. The process is shown in Figure 3.4. This method allows for priority annotation of the points containing the most information for the training of the final model.

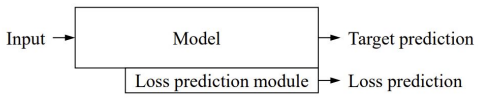


Figure 3.3: Addition of a module to an existing architecture to predict its error.[37]

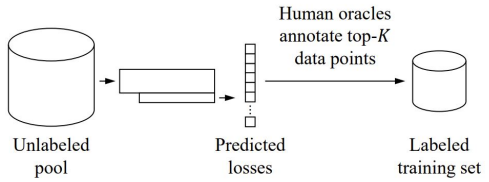


Figure 3.4: Iterative labelling process. The algorithm uses an oracle to annotate the K data points for which the module has predicted the largest error.[37]

Chapter 4

Methods

This chapter presents the methods used in this report. Section 4.1 presents the dose distribution prediction part starting with the architecture and data used (Sections 4.1.1 and 4.1.2) and ending with the parameters applied to train the models (Section 4.1.3). Then, three methods for uncertainty prediction are introduced: Monte-Carlo Dropout, bootstrap and a modification of the U-Net architecture in Sections 4.2 to 4.4. We conclude with the use of normalisation on uncertainty (Section 4.5) and the evaluation of our uncertainty measures in Section 4.6.

4.1 Dose prediction

4.1.1 Network Architecture

The architecture used in this thesis is U-Net (see Figure 3.1). Its general description has been made in Section 3.1. Instance normalisation[1] has been used to normalise the data, allowing faster neural network learning.

4.1.2 Data

To train our U-Net, we used a database of 200 head and neck tumour patients from the OpenKBP grand challenge (a competition to compare current dose distribution prediction methods [39]). In order to develop our methods, only the 200 training patients from the database for which the actual dose distribution was provided were used and then divided into 144 training, 36 validation and 20 testing patients.

[1] In the Instance Normalisation, mean and standard deviation are calculated for each individual channel and sample and across both spatial dimensions. In Batch Normalisation, the mean and standard deviation are calculated on each channel across all samples and both spatial dimensions.[38]

The patients were treated with intensity-modulated radiotherapy. For each patient, the database includes :

- the CT scan of the patient
- the possible dose mask (set of pixels where the dose can be non-zero)
- the mask of organs at risk among Brainstem, Spinal Cord, Right parotid, Left parotid, Esophagus, Larynx, Mandible
- the actual dose distribution
- the PTV prescription dose
- the voxel size
- probability matrices

Figure 4.1 shows the organs for which we have the mask for the 20 test patients, each with the mask for 4 to 7 organs.
All data are given as a NumPy matrix of (128x128x128) voxels of size around (3.5 mm$\times$3.5 mm$\times$2 mm).

Each patient's PTV is composed of 1 to 3 zones. The dose prescribed for each different area is defined by the severity of the tumour in that area: 70 Gy for gross disease (PTV70), 63 Gy for intermediate-risk target volumes (PTV63) and 56 Gy for elective target volumes (PTV56). Depending on the number of zones used for a patient, his prescription contains either : PTV70 (Figure 4.2(a)), PTV56 & PTV70 (Figure 4.2(b)) or PTV56 & PTV63 & PTV70 (Figure 4.2(c)). The different doses in the PTV prescription do not intersect and are such that they are no closer than 5mm to the patient's surface. The dose was delivered in 35 fractions by 9 equidistant coplanar beams (at 0° , 40° , . . ., 320°) with 6 MV step-and-shoot intensity-modulated radiation therapy.

Using data augmentation by random rotations, flips or translations is a common operation in deep learning. It allows to decrease overfitting and increase the size of the database. In our case, the data is augmented with dose distributions from synthetic plans generated via a variation of a published automated planning method by Babier et al. [40].

The probability matrices tell us what probability a row, column, slice should be chosen to further select the slices containing the PTV that is most important during the training.

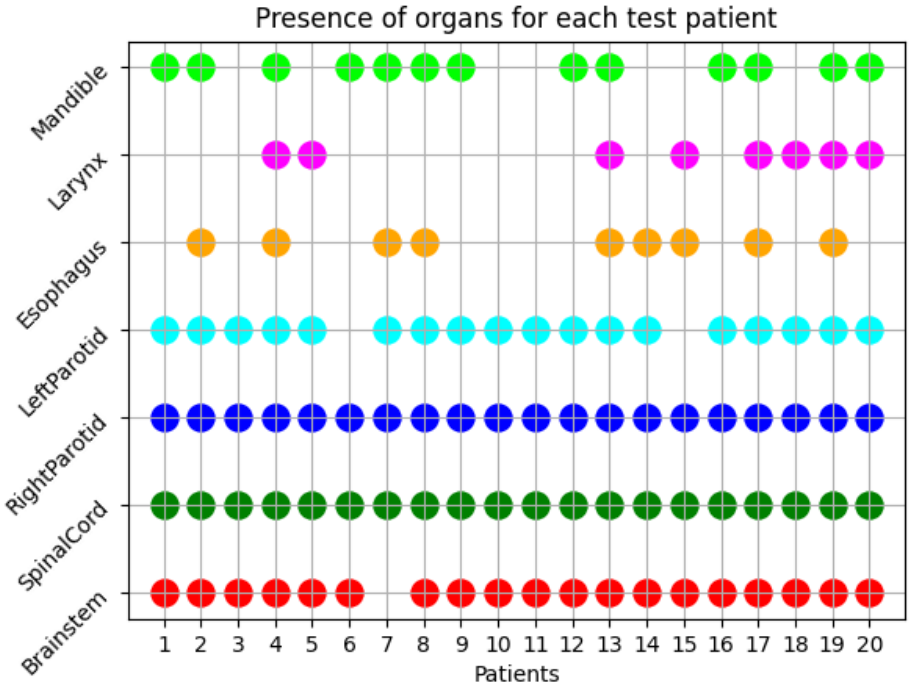
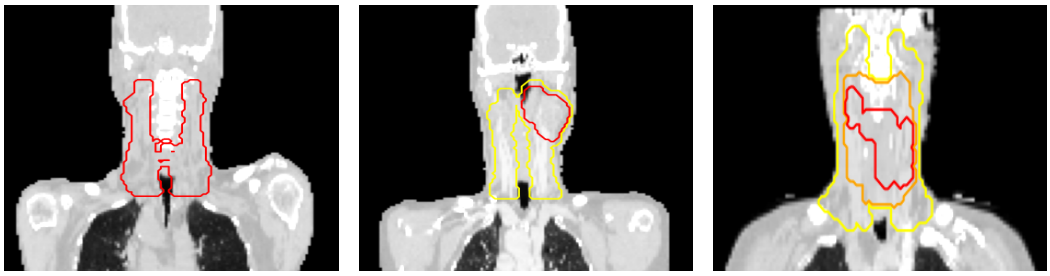


Figure 4.1: Organs for which we have the mask for the 20 test patients.



(a) (b) (c)

Figure 4.2: Example of patients with :
a) PTV70(red) **b)** PTV70(red) and PTV56(yellow)
c) PTV70(red), PTV63(orange) and PTV56(yellow)

4.1.3 Training

In this section, we will detail the parameters and methods used to train the models. Let us recall that the available patients were divided into 144 training, 36 validation

and 20 testing patients.

Batchsize

The batch size determines the number of examples provided to the model before doing a weight update during the network training. Due to the limitation of available memory on the GPU, the batch size used in this work is set to 1.

Epochs

An epoch corresponds to the forward and backward passage of the complete training set. For example, if the training set is divided into 500 batches, an epoch will be complete when all 500 batches have been processed once. The models in this work were trained over 100 epochs. When we look at the training history representing the training (blue) and validation (orange) loss over the 100 epochs (Figure 4.3), we observe a stabilisation of the validation loss, which is a sign that the training should be stopped. If we increase the number of epochs, the validation loss could rise, indicating overfitting.

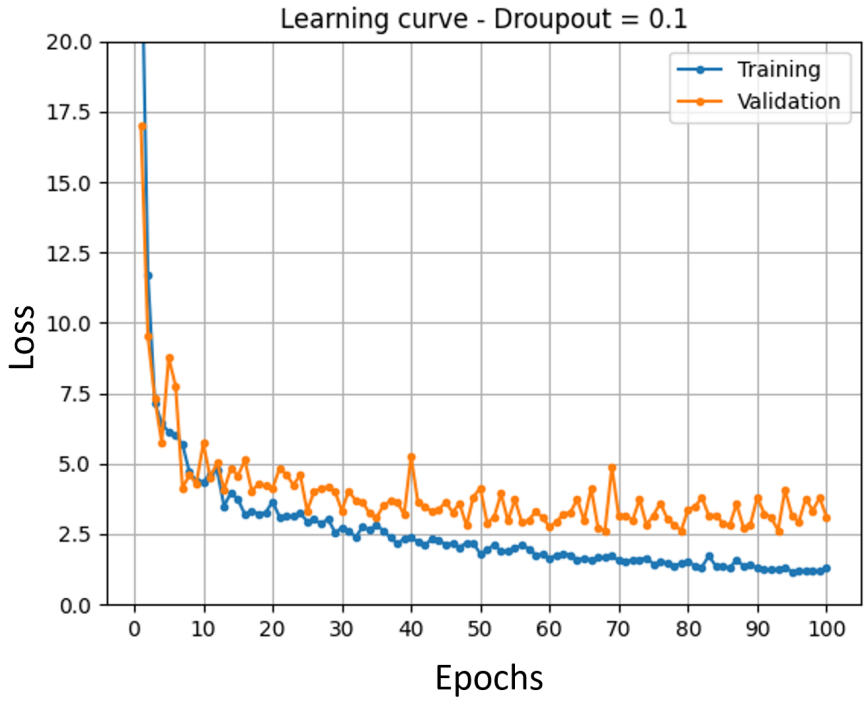


Figure 4.3: Learning curve representing the loss (MSE) on training (blue) and validation (orange) sets during training over 100 epochs.

Loss

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

The loss function (error that we want to minimise during training) used is the mean squared error (MSE). When we have $\hat{y}$ the prediction and y the actual value, the MSE is computed with formula (4.1) with a sum over voxels.

Optimizer

Gradient descent is among the most popular optimisation algorithms. It is an iterative process in which the parameters x of an objective function $f(x)$ are progressively updated to minimise it. The algorithm starts at point x_0. At iteration k+1, the gradient is calculated at the point x_k and a step is taken in the opposite direction to it, i.e. the steepest slope, to arrive at the point x_{k+1} such that $f(x_k) > f(x_{k+1})$ until reaching a minimum (which can be local if the function is non-convex).

$$x_{k+1} = x_k - \alpha \nabla f(x_k) \tag{4.2}$$

At each step, the parameters are updated according to the Formula (4.2), where α_k is called the learning rate and represents the step taken in the opposite direction to the gradient.

When the batch is equal to the entire training set in deep learning, it is called batch gradient descent. Batch gradient descent performs redundant calculations for large datasets, as it recalculates the gradient for similar examples before each parameter update. Stochastic gradient descent solves this problem by computing the gradient and updating the parameters after each training example (batchsize=1). Mini-batch gradient descent combines the two previous methods by updating the parameters based on a few training examples (1<batchsize<size of the training set).[41]

The main difficulty with the gradient descent algorithm is the choice of the learning rate. If the steps are too small, convergence will take too long. If too large, there is a risk of oscillating around the minimum without ever reaching it. To overcome this problem, the learning rate can be adapted over the iterations as proposed in many algorithms.

For this thesis, the Adam algorithm, which uses such an adaptive learning rate, has been used. Its name comes from adaptive moment estimation because it uses estimations of first and second moments of the gradient to adapt the learning rate

for each parameter (weight) of the neural network.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{4.3}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{4.4}$$

$$\hat{m} = \frac{m_t}{1 - \beta_1^t} \tag{4.5}$$

$$\hat{v} = \frac{v_t}{1 - \beta_2^t} \tag{4.6}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{4.7}$$

The first order moment estimate is given by the Equation (4.3) while the second order moment (uncentered variance) is given by Equation (4.4). m and v are initialized to zero, β_1 and β_2 have as default values 0.9 and 0.999 respectively, g is the current gradient. We must now make a bias correction with the Formulas (4.5) and (4.6). The weights (w) are finally updated with the rule from Equation (4.7).[42]

Patch-wise training

Training the model on the full matrices (128x128x128) would consume too much memory. To reduce the necessary computational capacity, sub-matrices are provided during training. Sub-matrices of size (96x96x96) are selected according to a probability distribution along each dimension (example in Figure 4.4 for a patient), favouring the slices most likely to contain the PTV. The different sub-matrices used are finally reassembled to form the final dose distribution.

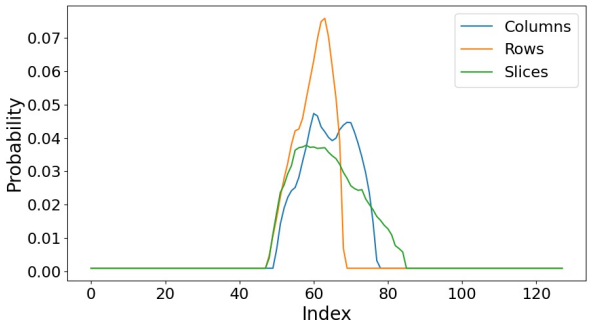


Figure 4.4: Probability density functions on rows, columns and slices to select a sub-matrix of size (96x96x96) used in the training of the neural network.

Computational power
A graphics process unit (GPU) is a processor designed for parallel processing. Originally used to accelerate 3D graphics rendering in video games, its use was later extended to deep learning. Neural networks are said to be *embarrassingly parallel*, meaning that the calculations performed require few efforts to be performed in parallel. The training of a deep neural network involves many calculations that can be performed independently (calculations on different neurons in the same layer). By taking advantage of parallelization (with a GPU), it can therefore be greatly accelerated.[43]

Four Nvidia GPUs were available to train models in this work: a GeForce GTX Titan with 6GB of memory, a Titan Xp and two Titan X Pascal each with 12GB of memory.

4.2 Monte Carlo dropout

Dropout is a technique that consists of ignoring a proportion $0 \leq p \leq 1$ of randomly chosen neurons in the network. Using a dropout when training the network avoids overfitting and forces the model to be more general (see Section 2.2.8). The dropout is commonly used during the training phase but is usually removed for testing.

The Monte Carlo dropout (MCD) (as seen in Section 3.2.1) is the first method presented in this thesis to obtain an uncertainty map. It has the great advantage of being adaptable to any neural network using dropout without modifying its architecture. The first step is to train a model using a dropout p (Training phase Figure 4.5). Then, we use this model to predict the dose distribution of our 20 test patients by applying the same dropout during testing. The dose distribution of each patient is predicted T times with the same model (Testing phase Figure 4.6). In our experiment, we trained five models with a dropout ranging from 0.1 to 0.5 and predicted T=100 times the dose distribution for each of the 20 patients. As dropout is applied during the test, each prediction will be made with a different model (some neurons in the model will be turned off), and the predictions will differ from each other.

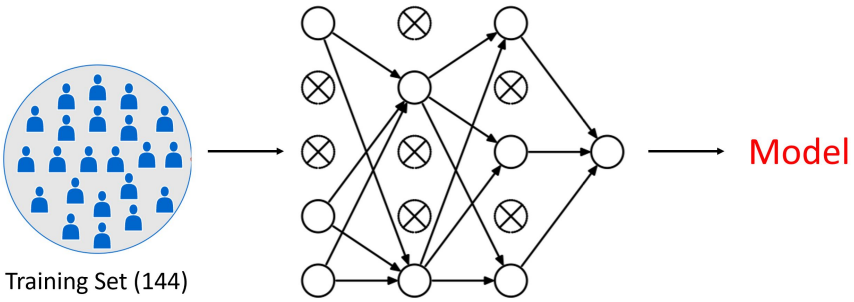


Figure 4.5: Training phase of Monte Carlo dropout method

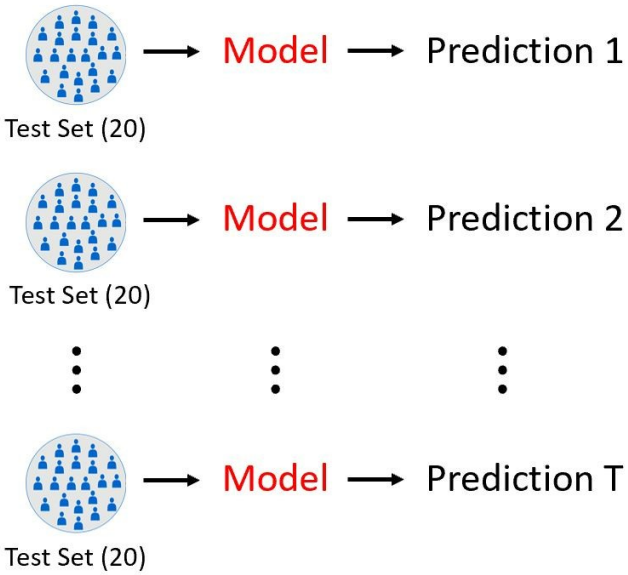


Figure 4.6: Testing phase of Monte Carlo dropout method

$$\mathbb{E}(y^*) \approx \frac{1}{T}\sum_{t=1}^{T}\hat{y}^*(x^*, W_t) \tag{4.8}$$

$$Uncertainty(y^*) \approx \sqrt{\frac{1}{T}\sum_{t=1}^{T}\hat{y}^*(x^*, W_t)\hat{y}^*(x^*, W_t) - \mathbb{E}(y^*)\mathbb{E}(y^*)} \tag{4.9}$$

We can now use the formulas proposed by [36] (reminded above) and already presented in Section 3.2.1. As a reminder, $y^*(x^*, W_t)$ represents the prediction of the model whose weights are W_t for an input x^*. In MCD, W_t represents the weights of the model when the dropout has been applied. The Formula 4.8 allows

us to calculate the expectation on the T predictions for each patient, which is then considered the model's general prediction. We can also calculate the standard deviation (SD) with Formula 4.9 on these predictions, which gives us a measure of the uncertainty. Indeed, a large SD tells us that the model can predict a very different dose distribution for the same input parameters, so it is not sure of its estimate. On the contrary, voxels with a very low SD reveal a model that always predicts the same dose for the given parameters, so it is considered confident about the answer.

4.3 Bootstrap

The second method used to find an uncertainty map is called bootstrap aggregation (bagging, already seen in Section 3.2.1). It is partly similar to MCD as it also uses SD as a measure of uncertainty. However, the dropout no longer plays a role in the uncertainty and so this method can be adapted to any neural network without any modification of the architecture. The bootstrap consists of training different models on a subset of patients randomly selected from the available training set. Therefore, a patient can be part of the training set of several models. Compared to MCD, the disadvantage of the bootstrap is the number of networks to train, one for the MCD on the whole training set against several for the bootstrap on subsets of the training set.

We trained T=20 models with a dropout of 0.1 to avoid overfitting and, for each, a training subset of 115 patients picked without replacement from the 144 available (Figure 2). We thus obtained 20 different models because they were trained on different training subsets. Then, during testing, we predicted once the dose distribution of each patient with each model. As for the MCD, we were able to make the expectation of the dose distribution and its SD on the T predictions with Formulas 4.8 and 4.9, which we consider as the final prediction of the model and its uncertainty. In these formulas, W_t represent the weights of each individually trained model.

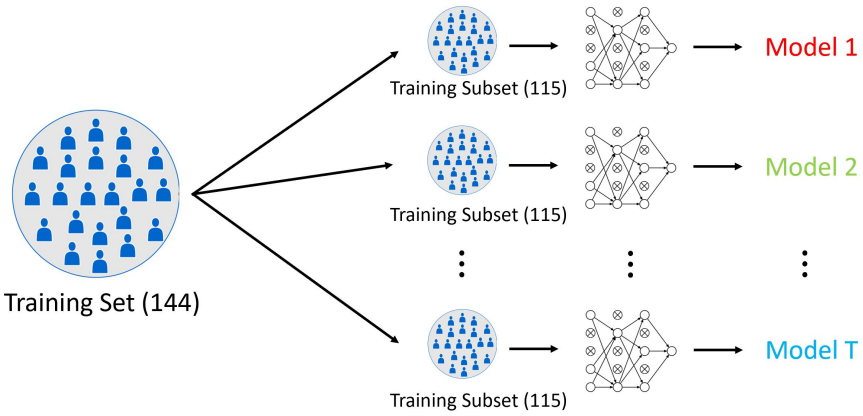


Figure 4.7: Training phase of boostrap method

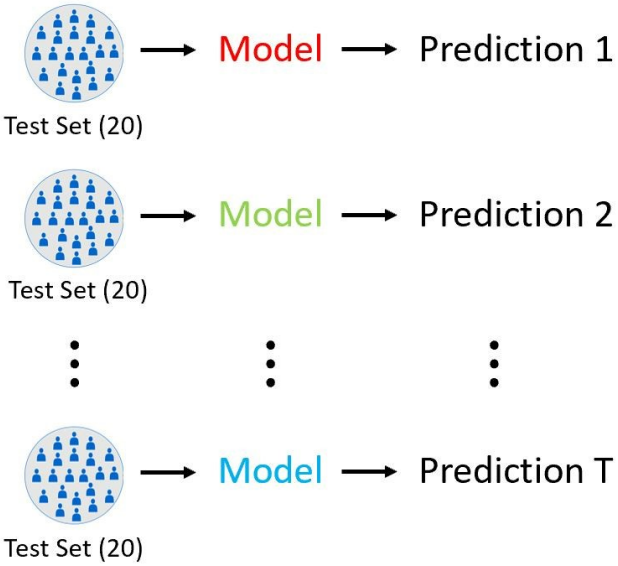


Figure 4.8: Testing phase of bootstrap method

4.4 Modified U-Net

The last method proposed in this thesis consists of using two U-Net models in series. The first model is a classical U-Net predicting the dose distribution while the inputs and outputs of the second have been modified to predict the error made by the former.

First, a regular model with a dropout rate of 0.1 was trained ($model_{pred}$) on the 144 test and 36 validation patients. The dose distribution of all 200 patients was evaluated with this model $T = 50$ times, allowing us to compute the SD and the mean absolute error (MAE) the model commits when predicting the dose, i.e., the difference between the ground truth and the mean of the 50 predictions, for each patient.

Next, a U-Net with additional inputs and a modified output definition is trained to predict the error committed by $model_{pred}$ (Figure 4.9). The dose distribution prediction and the SD were added to the existing inputs (patient anatomical information). On the output side, the dose prediction was replaced by the actual error made by the $model_{pred}$. A new model ($model_{error}$) was trained with the modified U-Net on the basis of the 144 training and 36 validation patients for whom the dose and standard deviation were previously calculated and was finally tested on the 20 test patients. The prediction of the error by the $model_{error}$ is considered as a measure of the uncertainty of the $model_{pred}$.

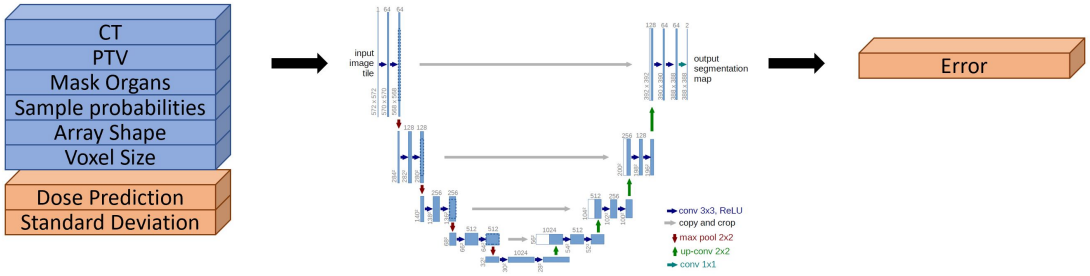


Figure 4.9: Modification of the U-Net, addition of prediction and standard deviation in input, prediction of error in output

4.5 Normalisation

For each of the three previous methods, we obtain an uncertainty map. We can normalise this by dividing each pixel of the uncertainty map by the value of the dose distribution at that pixel. This allows us to identify the points where the proportion of error to dose is most significant. The impact of an error of a few Gy would not have the same consequences on an organ at risk or on the tumour. Even if minimal, excess irradiation of OAR can lead to adverse effects due to their increased sensitivity.

Three normalisation methods were investigated:

The first one is to divide the uncertainty and error by the actual dose distribution at each voxel.

In a more realistic situation, the actual dose distribution is not available, in which case we can replace it with a prediction. When the uncertainty comes from the MCD method, the prediction used will come from the model trained with the same dropout. When the uncertainty comes from the bootstrap method, the bootstrap prediction will be used. Finally, when the uncertainty comes from the modified U-Net, the prediction used is the one from the model trained with $model_{pred}$.

Finally, in an even more concrete situation, the uncertainty is divided by a prediction of the dose distribution in order to guide the doctors to the areas where the error is most likely to occur and the error is divided by the actual dose distribution corresponding to the actual proportion of error made.

4.6 Uncertainty Evaluation

$$C = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{4.10}$$

The obtained uncertainty maps were evaluated by measuring their correlation with the error made in dose distribution prediction. The correlation was evaluated without normalisation and with each of the three normalisations mentioned above. The correlation is calculated using the Formula 4.10 where x_i is the value of voxel i in the uncertainty map and y_i is the error at the same voxel, $\bar{x}$ and $\bar{y}$ are the mean over x_i and y_i.

Chapter 5

Results

The results obtained for the prediction of the dose distribution (difference between the predicted and the actual dose distribution) are presented in the Section 5.1. The results for the measure of uncertainty are detailed in Section 5.2. All the results will be interpreted in the following chapter.

The values shown are generally averaged over all test patients. When the value applies to an OAR, the average is calculated on test patients who possess this organ mask without considering the others.

For clarity, we will use the following abbreviations: ADD when referring to the actual dose distribution and PDD when referring to a prediction of the dose distribution by a model.

$$\mathrm{MAE}(x, y) = \frac{1}{N} \sum_{i=1}^{N} |x_i - y_i| \tag{5.1}$$

The mean absolute error between two sets, x and y, is calculated according to Formula 5.1.

5.1 Dose prediction

5.1.1 MAE on DVH metrics: all models

First, the Table in Figure 5.1 shows the MAEs committed by the different models on the metrics D_{mean} (all organs), D_{95} and D_{max} (PTV) and D_2 (brainstem and spinal cord). The values shown are the mean and standard deviation (SD) over the 20 test patients. For each metric, the model with the best performance is shown in

green and the worst performance in red.

	PTV/OAR	Dropout = 0.1	Dropout = 0.2	Dropout = 0.3	Dropout = 0.4	Dropout = 0.5	Bootstrap
Dmean	Brainstem	1.19 ± 1.64	1.14 ± 1.2	1.28 ± 1.58	1.32 ± 1.41	1.21 ± 1.45	1.19 ± 1.39
	SpinalCord	0.98 ± 0.92	1.13 ± 0.91	1.24 ± 1.04	1.13 ± 0.99	1.19 ± 1.2	0.98 ± 0.98
	Right Parotid	2.07 ± 1.5	2.08 ± 1.26	1.88 ± 1.17	1.35 ± 1.01	2.13 ± 1.41	1.46 ± 0.93
	Left Parotid	1.46 ± 1.21	1.81 ± 1.17	1.51 ± 1.16	1.22 ± 0.77	1.26 ± 0.89	0.88 ± 0.64
	Esophagus	1.42 ± 1.16	1.44 ± 0.83	0.93 ± 0.66	1.05 ± 0.82	0.89 ± 0.84	0.92 ± 0.8
	Larynx	1.78 ± 1.67	2.04 ± 1.9	1.67 ± 1.09	1.84 ± 1.98	2.98 ± 2.4	1.6 ± 1.36
	Mandible	2.05 ± 1.37	1.92 ± 1.38	1.61 ± 0.97	1.43 ± 0.91	2.31 ± 1.95	1.42 ± 0.79
D95	PTV56	1.45 ± 1.25	1.52 ± 1.39	1.1 ± 1.06	1.54 ± 0.99	1.7 ± 1.2	1.39 ± 1.05
	PTV63	1.73 ± 1.89	1.98 ± 1.86	1.91 ± 1.56	1.69 ± 1.92	1.93 ± 1.98	1.63 ± 1.87
	PTV70	1.68 ± 1.02	1.75 ± 1.46	3.03 ± 1.28	1.37 ± 1.0	2.18 ± 1.37	1.5 ± 0.93
Dmax	PTV56	1.5 ± 1.05	1.73 ± 1.34	1.85 ± 1.3	1.69 ± 1.08	1.11 ± 0.94	1.62 ± 0.93
	PTV63	1.63 ± 1.19	2.17 ± 1.35	2.04 ± 1.29	2.27 ± 1.19	0.96 ± 0.64	1.83 ± 1.42
	PTV70	2.17 ± 1.89	2.55 ± 1.53	2.19 ± 1.01	2.69 ± 1.27	2.32 ± 2.31	2.08 ± 0.96
D2	Brainstem	2.99 ± 2.48	2.22 ± 1.87	2.52 ± 2.17	3.14 ± 2.26	2.44 ± 2.09	2.49 ± 1.86
	SpinalCord	1.9 ± 1.51	1.85 ± 1.28	1.93 ± 1.19	2.01 ± 1.38	3.06 ± 2.22	1.82 ± 1.43

Legend : Best result Worst result

Figure 5.1: MAE on the metrics for the different trained models: trained models with a dropout of 0.1 to 0.5 and bootstrap method. Values are reported as mean ± standard deviation on the 20 test patients. The best model is shown in green and the worst in red for each metric.

5.1.2 MAE on DVH metrics: Dropout 0.1 and Bootstrap

Figures 5.2 and 5.3 illustrate the MAE on D_{mean} and D_{max} for the different OAR and PTV prescriptions. Figures 5.4 and 5.5 represent the MAE on D_{95} and D_{99} on each possible PTV prescription dose (56, 63 and 70 Gy). Figures 5.6 and 5.7 show the MAE on D_2 and D_5 for the OAR.
The MAE is calculated for a trained model with a dropout of 0.1 in blue and the bootstrap model in orange. The red dots represent the median over the test patients and the grey bar connects the quantiles 0.25 and 0.75.
Figures 5.8 and 5.9 illustrate the DVH for the bootstrap and dropout of 0.1 models respectively.

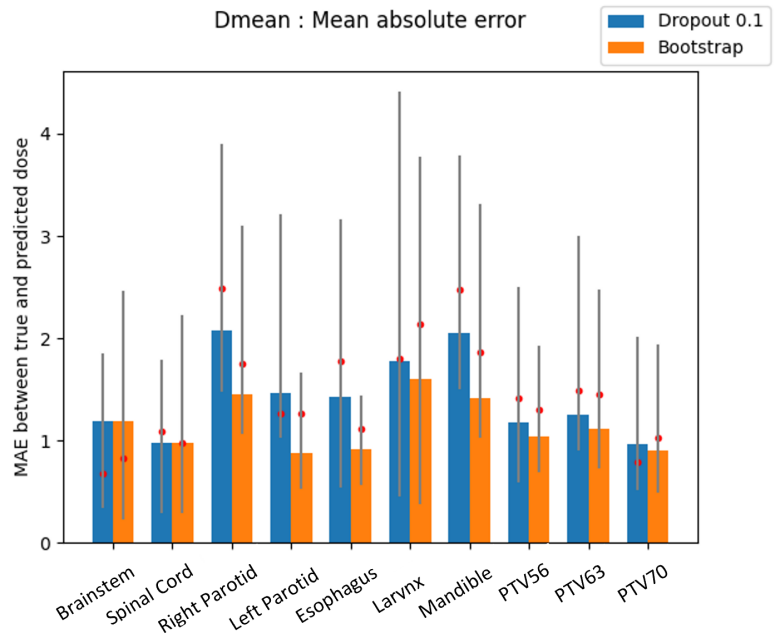


Figure 5.2: MAE [Gy] on D_{mean} for OAR and PTV prescriptions. Red dot is the median, grey bar connects quantiles 0.25 and 0.75.

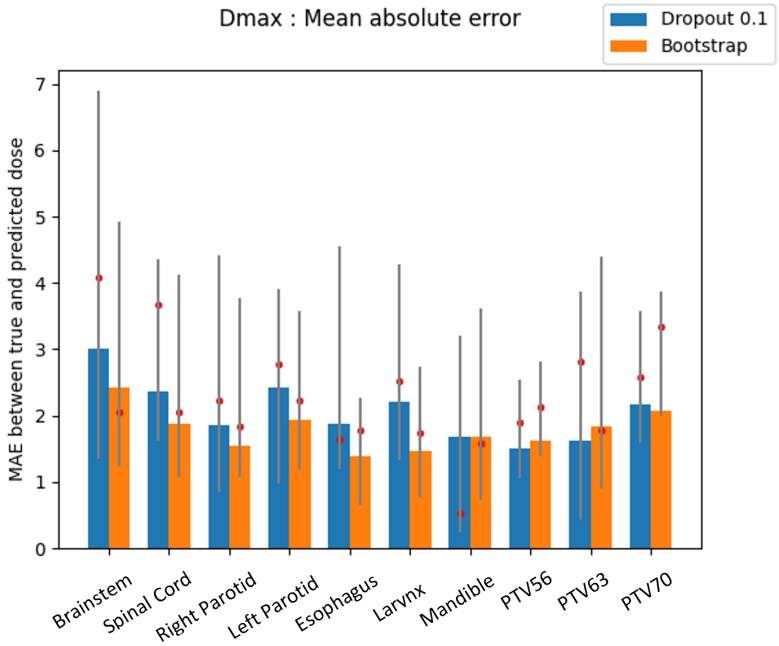


Figure 5.3: MAE [Gy] on D_{max} for OAR and PTV prescriptions. Red dot is the median, grey bar connects quantiles 0.25 and 0.75.

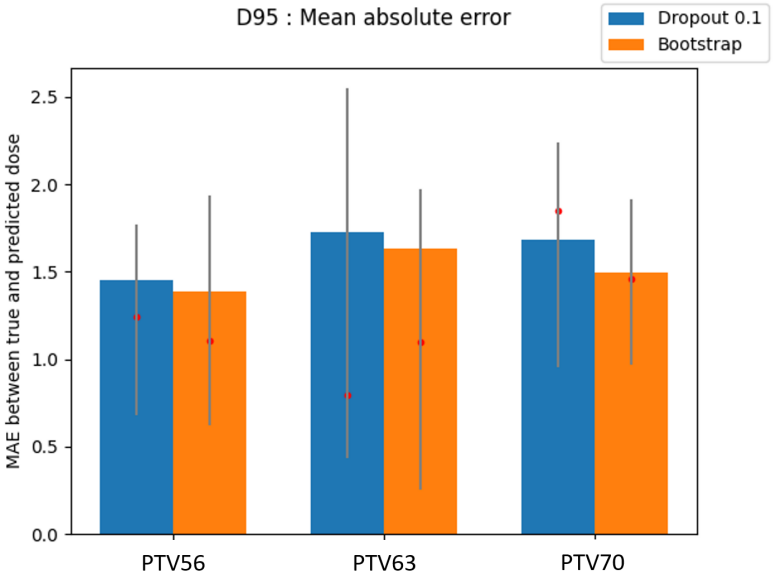


Figure 5.4: MAE [Gy] on D_{95} for PTV prescriptions. Red dot is the median, grey bar connects quantiles 0.25 and 0.75.

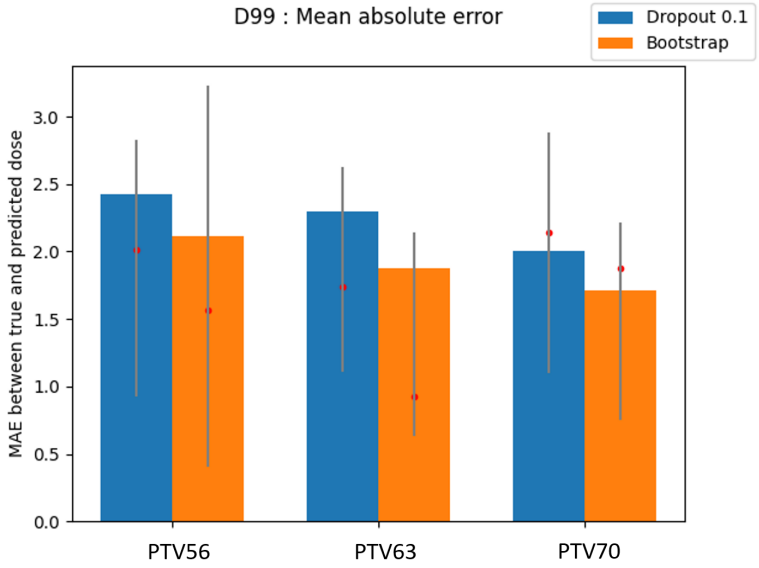


Figure 5.5: MAE [Gy] on D_{99} for PTV prescriptions. Red dot is the median, grey bar connects quantiles 0.25 and 0.75.

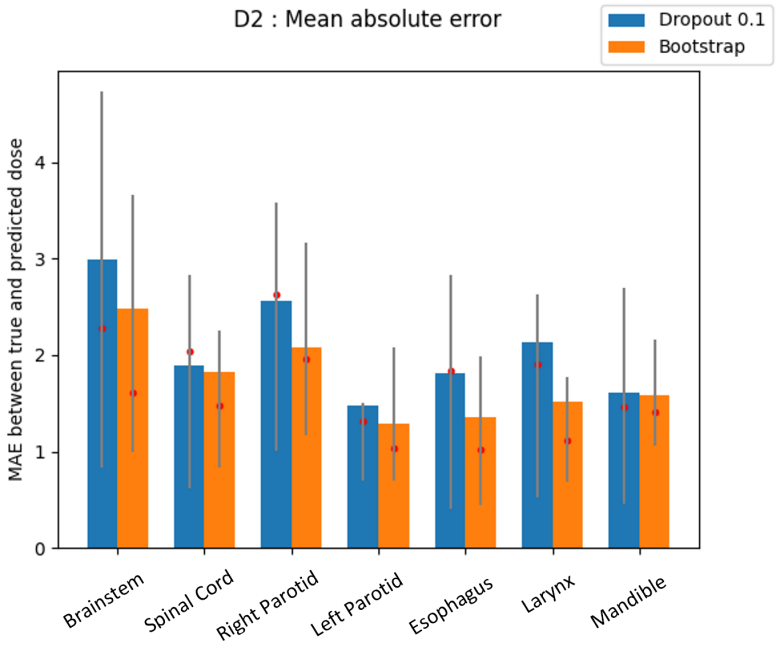


Figure 5.6: MAE [Gy] on D_2 for OAR. Red dot is the median, grey bar connects quantiles 0.25 and 0.75.

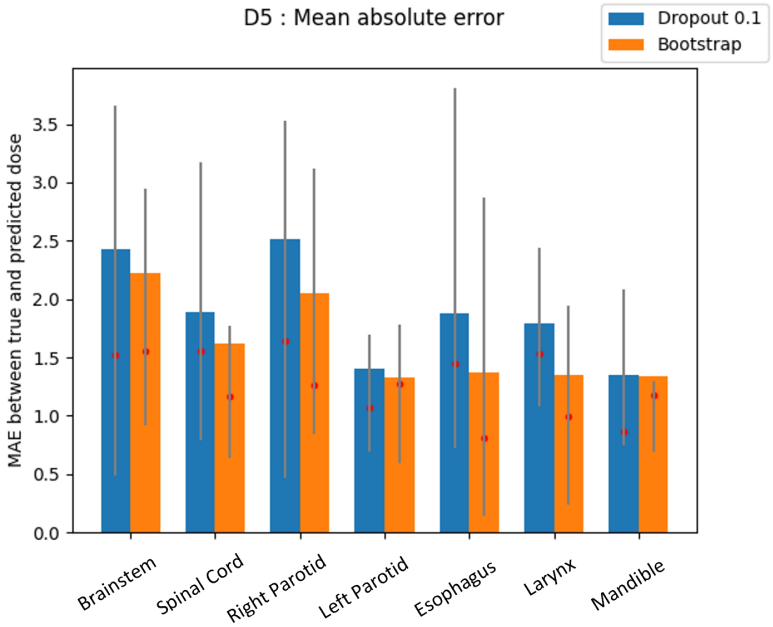


Figure 5.7: MAE [Gy] on D_5 for OAR. Red dot is the median, grey bar connects quantiles 0.25 and 0.75.

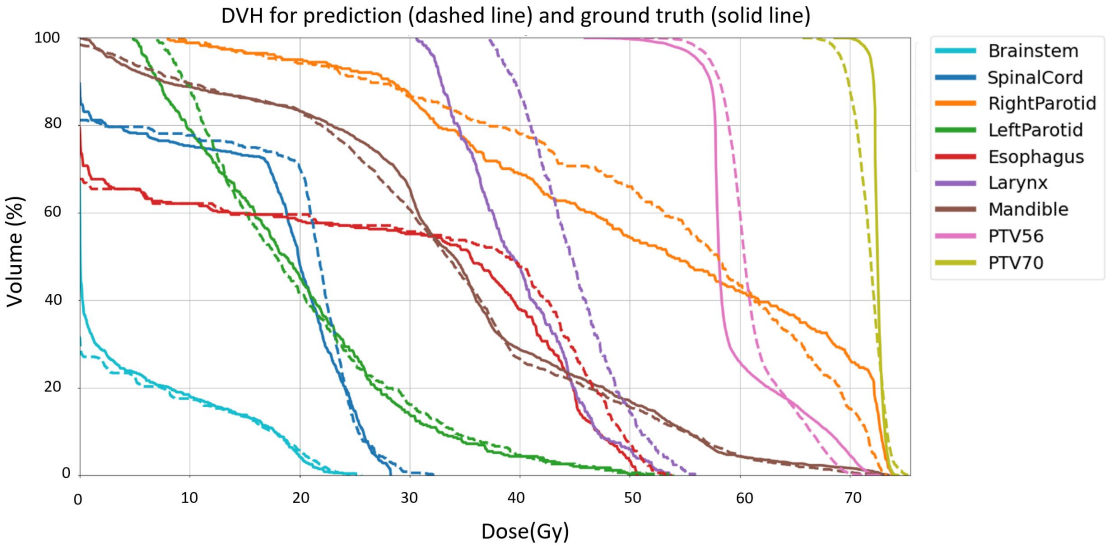


Figure 5.8: Dose Volume Histogram of patient 19 obtained with bootstrap model.

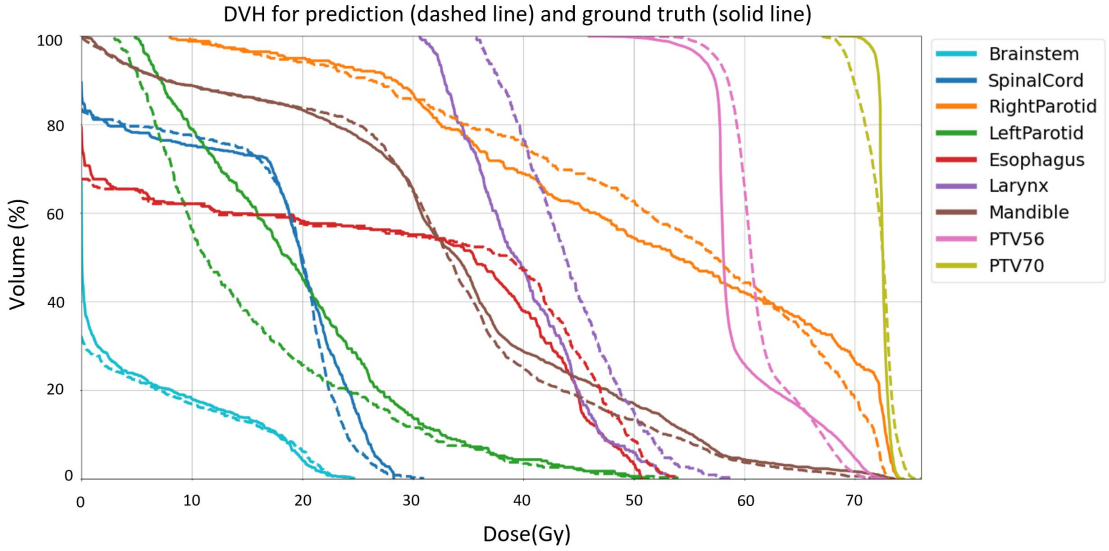


Figure 5.9: Dose Volume Histogram of patient 19 obtained with a model trained with a dropout of 0.1.

5.1.3 Visual comparison of prediction and ground truth

Figure 5.10 shows the predicted dose distribution (PDD), the actual dose distribution (ADD) and the MAE between them for the bootstrap method (above) and the 0.1 dropout (below).

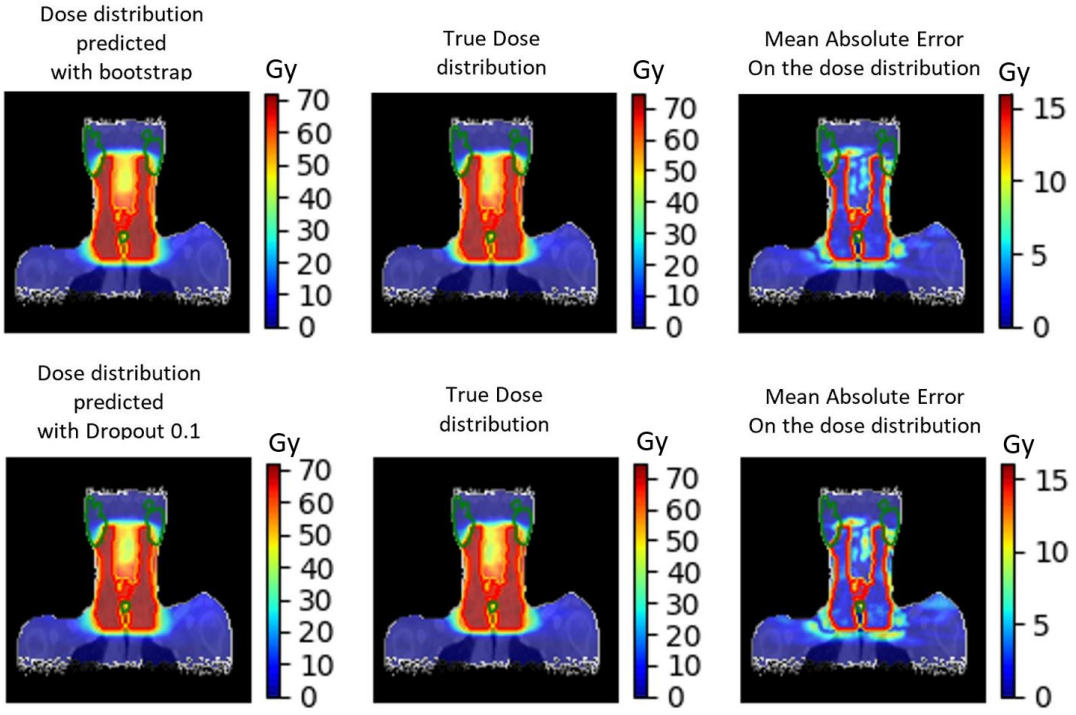


Figure 5.10: **From left to right** : Dose distribution prediction, actual dose distribution and mean absolute error between the prediction and the ground truth. **Above** the bootstrap method and **below** the model trained with a dropout of 0.1. The contour of the 70 Gy PTV prescription is drawn in red, contour of organs at risk(Right and left parotid, mandible and a part of the esophagus) are drawn in green.

5.2 Uncertainty

In this section, the results presented deal mainly with the correlation (see formula in Section 4.6) between a measure of uncertainty obtained by the methods detailed in Chapter 4 and MAE. We assume that the MAE throughout this section refers to the MAE calculated between the actual dose distribution (ADD) and its prediction (PDD) with Formula 5.1.

5.2.1 Uncertainty without normalisation

Figure 5.11 shows, for each uncertainty computation method, the average correlation between the uncertainty and the MAE for each OAR and the PTV.

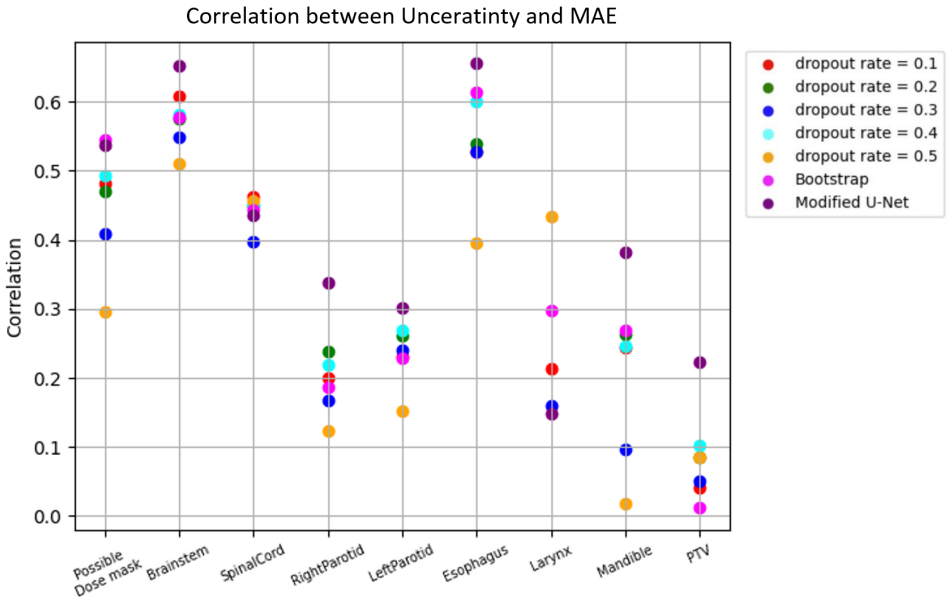


Figure 5.11: Correlation between uncertainty and prediction error(MAE) on OARs and PTV for the different presented methods.

Figure 5.12 indicates the mean correlation found on PTV and outside the PTV(Body-PTV) for each method.

	PTV	Body-PTV
dropout rate 0.1	0.04	0.454
dropout rate 0.2	0.08	0.448
dropout rate 0.3	0.05	0.392
dropout rate 0.4	0.103	0.466
dropout rate 0.5	0.085	0.323
Bootstrap	0.017	**0.513**
Modified U-Net	**0.223**	0.501

Figure 5.12: Correlation between uncertainty and MAE on PTV and outside (Body-PTV).

On Figures 5.13 and 5.14, scatter plots can be seen where the point coordinates are the uncertainty (x-axis) and MAE (y-axis) for the OAR (Figure 5.13) and PTV (Figure 5.14).

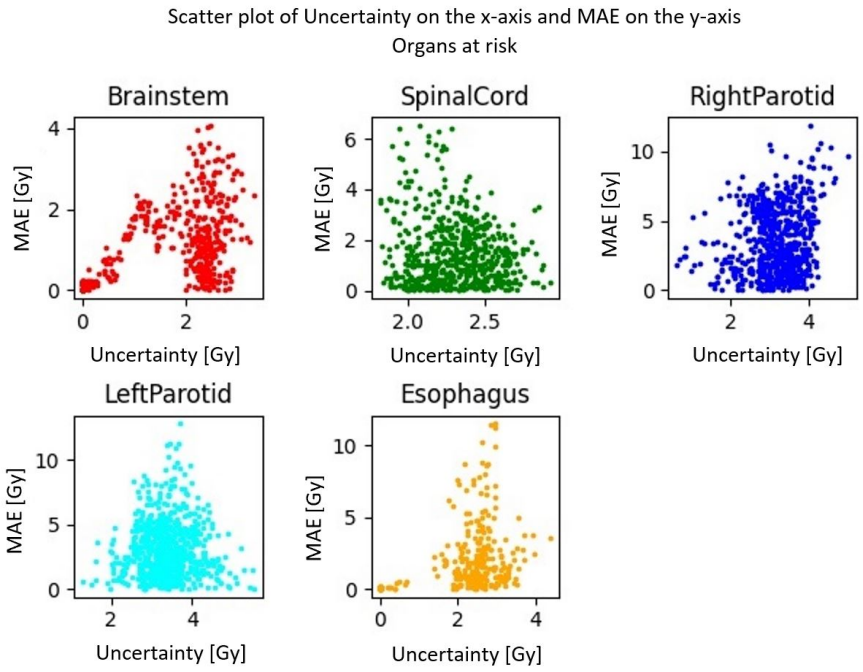


Figure 5.13: Scatter plot where the point coordinates are the uncertainty and MAE for OAR.

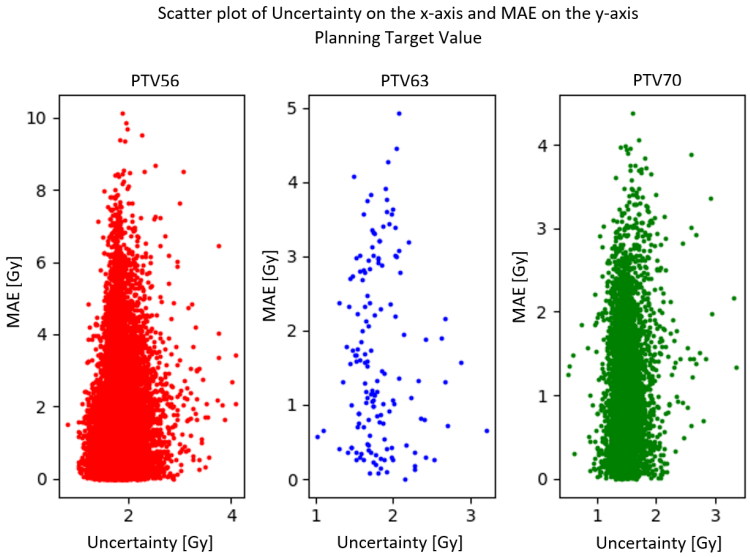


Figure 5.14: Scatter plot where the point coordinates are the uncertainty and MAE for PTV.

5.2.2 Uncertainty and error normalised by the actual dose distribution

In this part, we present the results found when the uncertainty map and the MAE are divided by the actual dose distribution.

The table presented at Figure 5.15 shows the correlation found between uncertainty and MAE maps on PTV and the rest of the body when they are both divided by the ADD. In order to avoid division by too small numbers and keep the results consistent, we only kept the points for which the ADD is >1 Gy. Larger correlations are observed when the limit is lowered.

	PTV	Body-PTV
dropout rate 0.1	0.115	0.659
dropout rate 0.2	0.159	0.669
dropout rate 0.3	0.021	0.686
dropout rate 0.4	0.154	0.704
dropout rate 0.5	0.215	0.255
Bootstrap	0.099	**0.714**
Modified U-Net	**0.27**	**0.714**

Figure 5.15: Correlations between uncertainty and MAE maps when divided by the actual dose distribution.

As in section 5.2.1, we made a scatterplot showing the uncertainty on the x-axis and the MAE on the y-axis normalised by the ADD. Only the points with a dose >0.1 Gy were kept in this case. Figure 5.16 focuses on OAR and Figure 5.17 on PTV.

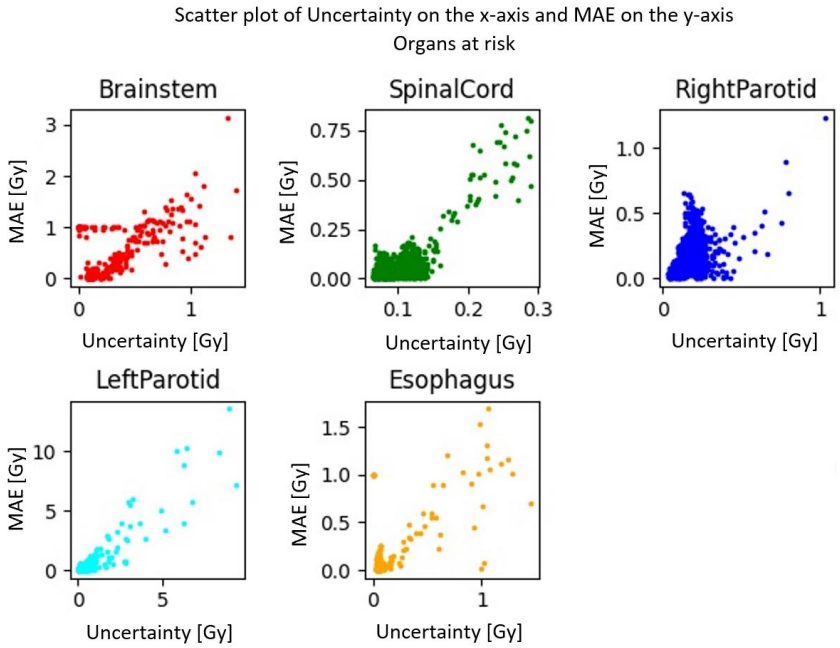


Figure 5.16: Scatter plot where the point coordinates are the uncertainty and MAE for OAR, both divided by the actual dose distribution. Only points where the dose is >0.1 Gy are kept.

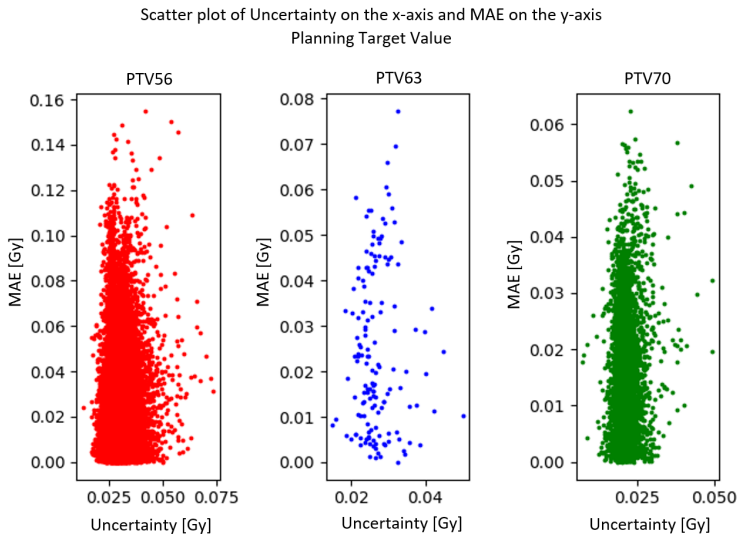


Figure 5.17: Scatter plot where the point coordinates are the uncertainty and MAE for PTV, both divided by the actual dose distribution. Only points where the dose is >0.1 Gy are kept.

5.2.3 Uncertainty and error normalised by a prediction of the dose

The same idea as in Section 5.2.2 is applied here but instead of dividing the uncertainty and error maps (MAE) by the actual dose distribution, we divide them by a prediction of the dose distribution. Only points were the PDD is >0.1 Gy are kept for the scatterplots and >1 Gy for pictures 5.18 to 5.21.

We first compare the correlations found for each patient on the possible dose mask when dividing by the ADD (green) or PDD (red) for the MCD of 0.1 (Figure 5.18), the bootstrap method (Figure 5.19) and the Modified U-Net method (Figure 5.20).

Figure 5.21 summarizes the correlations found on PTV and outside PTV between uncertainty and MAE divided by a PDD.

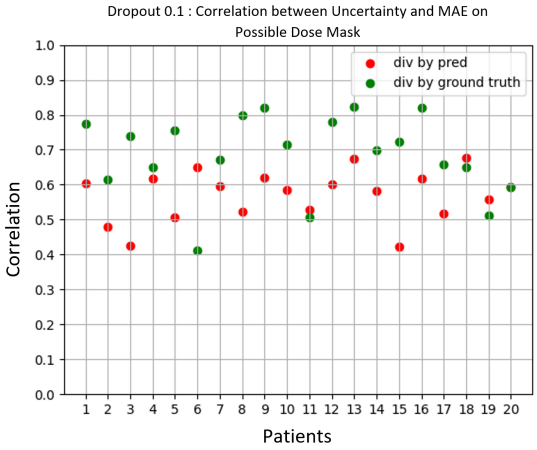


Figure 5.18: Correlation between uncertainty and MAE both divided by the actual dose distribution (green) and a prediction of the dose distribution (red) for each patient and the dropout 0.1 MC method.

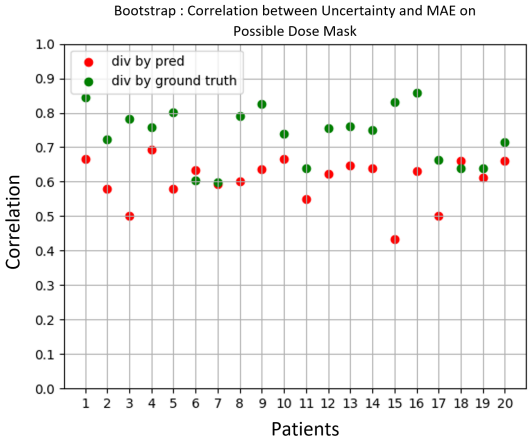


Figure 5.19: Correlation between uncertainty and MAE both divided by the actual dose distribution (green) and a prediction of the dose distribution (red) for each patient and the bootstrap method.

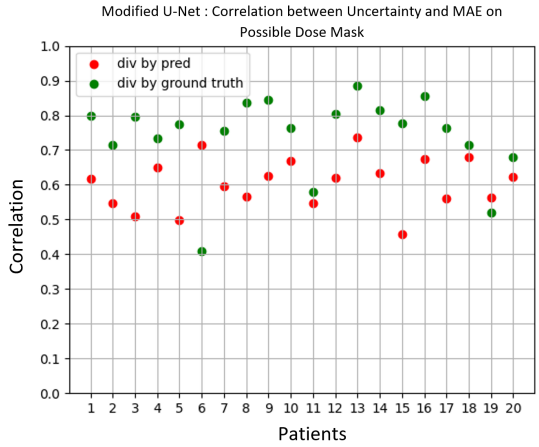


Figure 5.20: Correlation between uncertainty and MAE both divided by the actual dose distribution (green) and a prediction of the dose distribution (red) for each patient and the Modified U-Net method.

	PTV	Body-PTV
dropout rate 0.1	0.01	0.524
dropout rate 0.2	0.136	0.525
dropout rate 0.3	0.061	0.55
dropout rate 0.4	0.135	0.533
dropout rate 0.5	0.173	**0.624**
Bootstrap	0.085	0.559
Modified U-Net	**0.274**	0.558

Figure 5.21: Correlations between uncertainty and MAE when both are divided by a prediction of the dose distribution

Scatterplots representing the uncertainty on the x-axis and the MAE on the y-axis normalised by a PDD are shown in Figures 5.22 (OAR) and 5.23 (PTV).

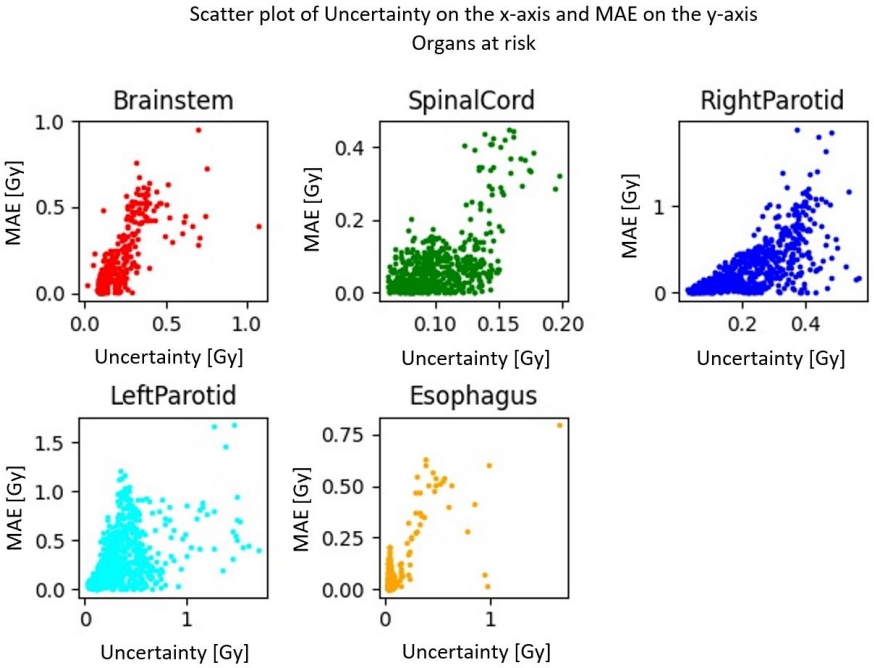


Figure 5.22: Scatter plot where the point coordinates are the uncertainty and MAE for OAR, both divided by a prediction of the dose distribution. Only points where the PDD is >0.1 Gy are kept.

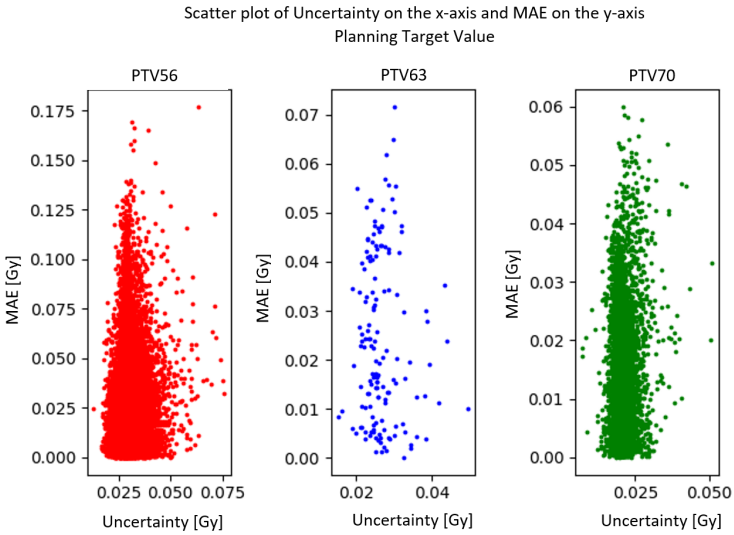


Figure 5.23: Scatter plot where the point coordinates are the uncertainty and MAE for PTV, both divided by a prediction of the dose distribution. Only points where the PDD is >0.1 Gy are kept.

5.2.4 Visual comparison of Modified U-Net uncertainty with different normalisations

Figure 5.24 shows the uncertainty map (left), the MAE (centre) and the actual dose distribution (right) for the same patient. Uncertainty and MAE are not normalised at the top, normalised by the actual dose distribution in the middle and normalised by a dose distribution prediction at the bottom. The Modified U-Net method was used in the three cases to obtain a map of the uncertainty. For the uncertainty maps, the points with the slightest uncertainty have been removed to highlight the points with the most significant uncertainty aligned with the MAE. For this figure, we kept the points whose uncertainty is greater than the 0.5 quantile (i.e. half of the points were kept).

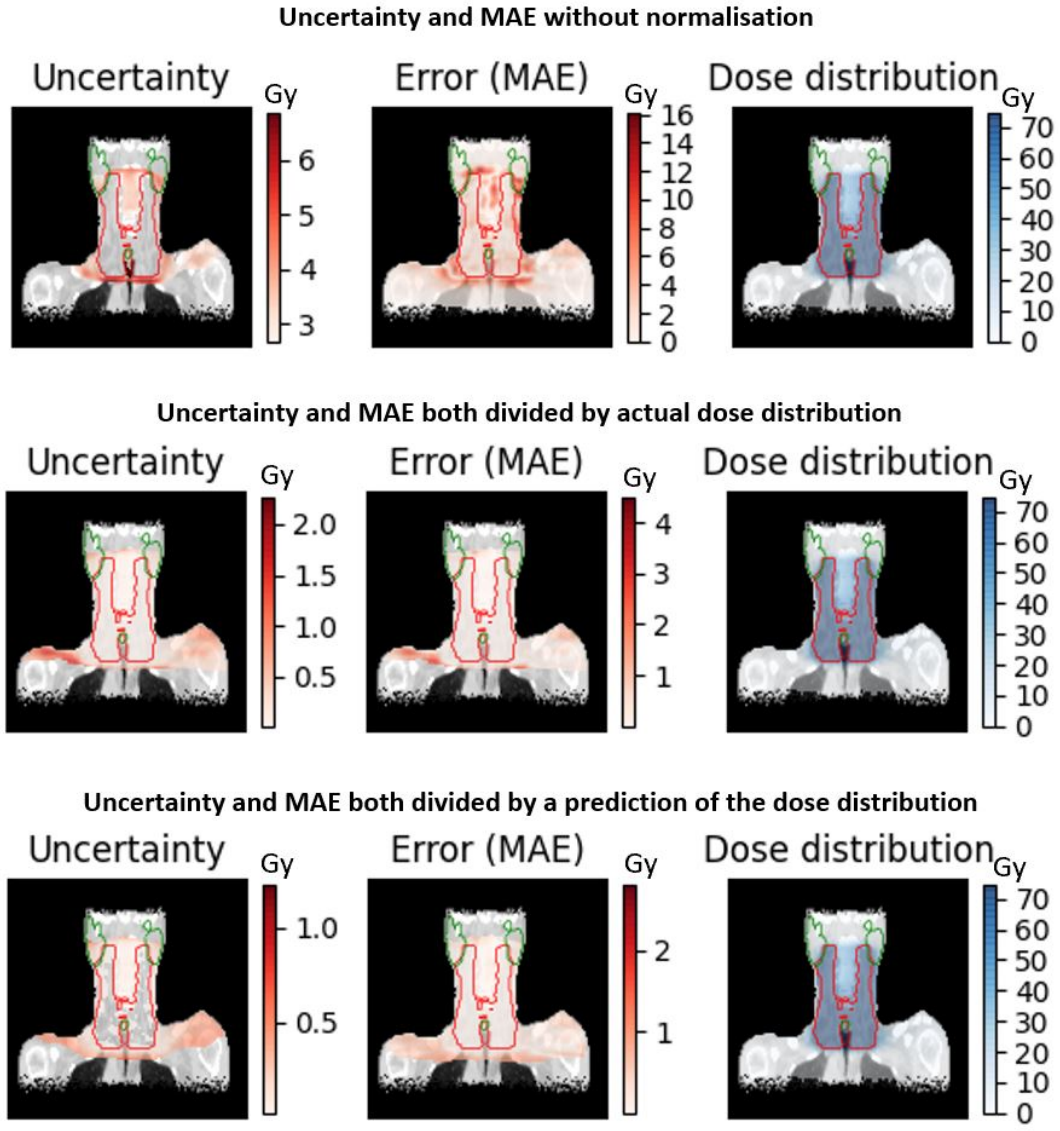


Figure 5.24: **From left to right :** Uncertainty map, MAE, Actual dose distribution.
From top to bottom : Modified U-Net without normalisation, nomalised by the actual dose, normalised by a prediction of the dose.

5.2.5 Uncertainty and error normalised by a prediction of the dose and the actual dose respectively

We can also divide the uncertainty by a PDD and the error by the ADD, thus obtaining a correlation on and outside the PTV as shown in Figure 5.25.

	PTV	Body-PTV
dropout rate 0.1	0.105	0.345
dropout rate 0.2	0.131	0.329
dropout rate 0.3	0.07	0.302
dropout rate 0.4	0.136	0.3
dropout rate 0.5	0.167	**0.447**
Bootstrap	0.087	0.348
Modified U-Net	**0.277**	0.379

Figure 5.25: Correlations between uncertainty and MAE when divided by a prediction of and the actual dose distribution respectively.

5.3 Computational time

Training a model with a dropout of 0.1 on 144 test patients and 36 validation patients took 6h35min. For the bootstrap method, each model trained on the 115 test patients (and 36 validation patients) required 5h30min of training.

The evaluation of the dose distribution of a patient takes about 8 seconds. The bootstrap method requires the evaluation of the dose distribution for 20 models and takes about 2min45sec.

Chapter 6

Discussion

6.1 Dose distribution prediction

Comparison of MCD with different dropout rates

First of all, we notice from the table of Figure 5.1 that the model trained with a dropout of 0.1 is globally better than with the other dropouts. It is the best model (excluding bootstrap) for the D_{mean} on the Spinal Cord and D_{max} on the PTV70. However, unlike the other dropouts, it is never the worst, which makes it a good general model. The more unsatisfactory results with increasing dropout could be explained by the fact that dropout increases the generality of a model. When too high a dropout is applied, too many neurons are turned off, and the model can no longer learn in a sufficiently complex way.

Comparison of Monte Carlo and bootstrap methods

From the same table (Figure 5.1), one can conclude that the bootstrap method is the best-proposed method for dose distribution prediction. It obtains a lower MAE than the MCD 0.1 for all the metrics shown except for D_{max} on PTV56 and 63 with, despite this, a low error of 2.28% (PTV56) and 2.46% (PTV63) of the real value.

On the PTV, the difference between the MCD 0.1 and the bootstrap is significant (mostly in favour of the bootstrap) and does not exceed 0.43 Gy on D_{99} and 0.2 Gy on D_{95}, D_{mean} and D_{max}.

On the OAR, a clear difference is present on D_{max}, D_2 and D_5 (up to a difference $\geq$0.5 Gy) except for the mandible, where both methods have relatively similar

results. On D_{mean}, differences are present (up to 0.63 Gy in favour of bootstrap) on most OARs except for the brainstem and spinal cord, where the error is similar. The lack of difference between the two models can be explained by the presence of these organs in most training patients (see Figure 4.1). In contrast, the larynx and esophagus, for which there is a difference between the two models, are not present in many patients. The bootstrap method would therefore have a better learning capacity on structures with few examples. A final observation is that the errors made by the bootstrap are less variable according to the OARs studied than the MCD 0.1, so it can be considered more stable.

We also find the best bootstrap results on the DVH (Figures 5.8 and 5.9) with a clear difference on the left parotid (green) and a slight difference for the brainstem (cyan) and larynx (purple). In Figure 5.10, a slightly smaller bootstrap error can be observed in some areas, such as the bottom and the centre of the MAE (right figures).

6.2 Uncertainty maps

Without normalisation, Figure 5.11 shows a better correlation on the majority of organs with the modified U-Net. On the possible dose mask, the bootstrap and the modified U-Net methods are almost equivalent with correlations of respectively 0.513 and 0.5 on the Body-PTV. On the PTV, the modified U-Net is far better than the other models.

We see a significant increase in the correlation due to the dose distribution normalisation, which becomes >0.65 for all methods except for the 0.5 dropout (Figure 5.15). Although this result is only theoretical because the ground truth dose distribution is not available for a new patient outside our database, it indicates an interesting relationship between the uncertainty we can obtain and the actual error. The scatter plots on the OARs are also indicative of this correlation. Indeed, Figure 5.16 (without normalisation) shows much more scattered points than Figure 5.13 (with normalisation). On the PTV, the correlation increases only slightly (+0.047 with the modified U-Net) and the scatter plots are very similar before and after normalisation (Figures 5.14 and 5.17).

Normalisation by a predicted dose distribution allows to get closer to a real applicative case without significant impact on the PTV correlation and keeping a correlation around 0.56 (or even 0.6 with the 0.5 dropout) outside the PTV. Figures 5.18 to 5.20 show the impact of using a prediction rather than the actual distribution dose. A general decrease in correlation which may, however, be very

low for some test patients. The uncertainty measure can still be used directly or as part of a more complex algorithm. The scatter plot of OARs (Figure 5.22) shows less dispersion than without normalisation but still more than with normalisation by the actual dose distribution.

Visually, Figure 5.24 reveals a high degree of similarity between the uncertainty and the MAE after normalisation. There is also a similarity that is less precise but still present before normalisation. It can be seen in this figure that the uncertainty becomes larger at lower doses and is very small on the PTV. The error follows the same pattern, which is a positive point for the uncertainty measure. This behaviour was observed in a general manner for the other patients as well. Let us recall that the lowest values of uncertainty were discarded from the image for a better comparison. Another general observation is that the uncertainty prediction is smoother than the MAE which contains large error spots. Still on the same figure, one can notice the organs near the PTV (in green) that need special attention during treatment planning. A large error is present on their lower part, also detected by the uncertainty measurement. On these OARs, the calculated uncertainty would indeed allow us to indicate the error areas to the dosimetrist, who could correct them manually.

The last part of the results presents the correlation between the uncertainty normalised by a dose distribution prediction and the MAE normalised by the ground truth dose distribution. Unsurprisingly, the correlation is weaker than in the previous cases because the normalisation is different on the uncertainty and MAE. Nevertheless, a correlation up to 0.45 is observed, indicating a moderate link between the two measures.

In general, the correlation is always weaker on the PTV than outside. This can be explained by the fact that the error there is much smaller in proportion to the dose received (typically between 50 and 80 Gy), so the model uncertainty is less easily detected.

6.3 Our work in litterature

This work is partly similar to that proposed by Nguyen et al. [36]. We can therefore compare the results obtained. Overall, the results related to dose distribution prediction are pretty similar.

For Dmean, the MAE on PTV is in both cases between 1 and 2% of the prescription (70 Gy). On the OAR, the differences between the two studies are minimal (<1%=0.07 Gy) except for the esophagus where our bootstrap model commits an MAE of about 1 Gy, i.e., less than 1.5% against an MAE slightly below 3% in [36].

For Dmax, the MAE on the PTV70 is slightly higher in our work (less than 1% of the prescription). Contrary to [36], in our case, the error on PTV70 is lower with the bootstrap method than with an MCD. On the OAR, the results are again similar with differences between the two models <1% of the prescription dose except on the esophagus where our study is slightly better (MAE about 1.5% smaller for the bootstrap).

On the D95, the difference between the two studies is never greater than 1% of the prescription (slight advantage for [36] on the PTV56).

On the D99, the advantage is marked on the PTV70 where we make an MAE almost two times smaller. [36] is slightly better on PTV56 and somewhat worse on PTV63 but still by a tiny percentage.

Concerning uncertainty, Nguyen et al. find a correlation of 0.55(Body-PTV) and 0.27(PTV) with the bootstrap and 0.47(Body-PTV) and 0.026(PTV) with the MCD using their scaling method (see section 3.2.1). We find similar correlations without normalisation with 0.513(Body-PTV) and 0.223(PTV) for the bootstrap and 0.45(Body-PTV) and 0.04(PTV) with the MCD. After normalisation by the actual dose distribution, we obtain much higher correlations (see Figure 5.15). When normalising by a prediction of the dose distribution, we get a correlation up to 0.62 on Body-PTV and 0.27 on PTV as [36].

The big difference is the post-processing applied to the uncertainty (normalisation here, scaling for [36]). The scaling method proposed by [36] requires more computation as the scaling factor has to be computed from the validation patients and over a region of interest. In our case, the dose distribution prediction is already available and facilitates the implementation of the method.

6.4 Further researches

In this work, we used deep learning to predict the optimal dose distribution. However, the plan does not indicate the technical parameters of the beams that allow obtaining this dose distribution effectively. Therefore, a step is still needed to make the treatment applicable, either by inverse planning or by dose mimicking, by staying as close as possible to the predicted dose distribution. A future study could investigate this step and the uncertainty linked to it as we have done in this text.

6.5 Possible application : Active learning with uncertainty

One possible application of uncertainty is to integrate it into an active learning algorithm. It will be possible to set a level of uncertainty above which the predicted distribution dose of a patient is considered sure enough. Using the algorithm detailed in Section 3.2.2, we can replace the loss module that predicts error with a measure of uncertainty. As long as points have an uncertainty above this threshold, we continue to iterate. This measure does not require implementing an appendix to the main architecture as in the method of [37] and can be used from any existing architecture. The use of uncertainty is, therefore, more flexible and easier to set up. A summary of the algorithm adapted to uncertainty is given below.

Initial data sets :
Set of unlabelled data (all data, size N) : $\mathcal{U}_N$
Set of labelled data (empty) : $\mathcal{L}_\emptyset$
Initialisation :
Pick K data points from $\mathcal{U}_N$ at random and label them
$\mathcal{U}_N \rightarrow \mathcal{U}^0_{N-K}$
$\mathcal{L}_\emptyset \rightarrow \mathcal{L}^0_K$
Iteration i :
Train the model with set $\mathcal{L}^{i-1}$ as the training set
Evaluate uncertainty of data points of $\mathcal{U}^{i-1}_{N-iK}$
Query labelling of the K most uncertain points
Add new labelled points to $\mathcal{L}$
$\mathcal{U}^{i-1}_{N-iK} \rightarrow \mathcal{U}^i_{N-(i+1)K}$
$\mathcal{L}^{i-1}_{iK} \rightarrow \mathcal{L}^i_{(i+1)K}$

Chapter 7

Conclusion

The automation of treatment planning is a currently popular topic on which many studies have been conducted and in which new technologies such as deep learning have a prominent role to play. Limiting the intervention of a specialist by automating the more general tasks allows him to use his knowledge optimally. By relieving the physician of time-consuming processes, he can focus on complex cases or, in general, improve the quality of the plans.

In this thesis, we compared different methods of dose distribution prediction, namely, the training of a deep neural network with several dropout rates and the bootstrap method. From our observations, we concluded that the method giving the best results was the bootstrap method. The MAE on the test patients does not exceed 2.5 Gy and is better than the other methods for almost half of the metrics studied.

Furthermore, we provided several techniques allowing the model to convey its uncertainty to the user in an easily interpretable way: the Monte-Carlo Dropout, the bootstrap and a modification of the U-Net architecture originally used to predict the dose distribution. We also observed a significant increase in the correlation between uncertainty and error by normalising both. The method giving the best overall results is the modified U-Net method. This gives a correlation of :

- without normalisation: 0.5 on the Body-PTV and 0.22 on the PTV
- with normalisation by the actual distribution dose: 0.71 on the Body-PTV and 0.27 on the PTV
- with normalisation by a dose distribution prediction: 0.56 on the Body-PTV and 0.27 on the PTV.

The result of this study can be used directly to be integrated into the treatment

planning following the dose distribution prediction. It can also be used as part of an active learning algorithm to optimise the data points annotation process with a limited budget.

Unfortunately, the optimal distribution dose predicted by the neural network is not always achievable. One perspective may be to compute the radiation parameters to be used when administering the treatment to the patient to get as close as possible to the prediction. In the same manner as done in this thesis, the measurement of uncertainty could be studied at this level of treatment planning.

Bibliography

[1] Cancer Today, Global Cancer Observatory, 2020, URL: https://gco.iarc.fr/today/online-analysis-pie?v=2020&mode=cancer&mode_population=continents&population=900&populations=900&key=total&sex=0&cancer=39&type=0&statistic=5&prevalence=0&population_group=0&ages_group%5B%5D=0&ages_group%5B%5D=17&nb_items=7&group_cancer=1&include_nmsc=1&include_nmsc_other=1&half_pie=0&donut=0.

[2] *Radiation Therapy to the Brain*, Memorial Sloan Kettering Cancer Center, 2018, URL: https://www.mskcc.org/cancer-care/patient-education/radiation-therapy-brain.

[3] M.D. Kris L. Gast, *Eclipse Treatment Planning System*, Fort Smith Radiation Oncology, P.A. URL: http://www.fsro.net/?page_id=768.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015, arXiv: 1505.04597 [cs.CV].

[5] Wikipedia contributors, *Gray (unit) — Wikipedia, The Free Encyclopedia*, [Online; accessed 8-April-2021], 2020, URL: https://en.wikipedia.org/w/index.php?title=Gray_(unit)&oldid=991253031.

[6] *File:Blausen 0205 CATScan 01.png*, Wikimedia Commons, 2021, URL: https://commons.wikimedia.org/wiki/File:Blausen_0205_CATScan_01.png.

[7] Qualimedis, URL: https://qualimedis.fr/produit/qfix-masque-thermoplastique-et-kevlar/.

[8] Ramona Charaghvandi et al., "Redefining radiotherapy for early-stage breast cancer with single dose ablative treatment: A study protocol", in: *BMC Cancer* 17 (Mar. 2017), p. 181, DOI: 10.1186/s12885-017-3144-5.

[9] Burnet NG et al., "Defining the tumour and target volumes for radiotherapy", in: *Cancer Imaging* 4(2) (2004), Published 2004 Oct 21, pp. 153–161, DOI: 10.1102/1470-7330.2004.0054.

[10] Raystation, RaySearchLaboratories, URL: https://www.raysearchlabs.com/raystation/.

[11] Dr Emami B, "Tolerance of Normal Tissue to Therapeutic Radiation", in: *Department of Radiation Oncology, Loyola University Medical Center, Maywood, Illinois, USA* 1 (Spring 2013), pp. 35–48.

[12] Marshall Hargrave, *Deep Learning*, Investopedia, 2020, URL: https://www.investopedia.com/terms/d/deep-learning.asp#:~:text=Deep%20learning%20is%20an%20artificial,for%20use%20in%20decision%20making.&text=Also%20known%20as%20deep%20neural%20learning%20or%20deep%20neural%20network..

[13] Jürgen Schmidhuber, "Deep learning in neural networks: An overview", in: *Neural Networks* 61 (Jan. 2015), pp. 85–117, ISSN: 0893-6080, DOI: 10.1016/j.neunet.2014.09.003, URL: http://dx.doi.org/10.1016/j.neunet.2014.09.003.

[14] Niji, *Introduction au Deep Learning*, slideshare, 2019, URL: https://www.slideshare.net/Niji_Digital/introduction-au-deep-learning.

[15] Vaibhav Sahu, *Power of a Single Neuron*, Towards Data Science, June 2018, URL: https://towardsdatascience.com/power-of-a-single-neuron-perceptron-c418ba445095.

[16] Vikram Mullachery, Aniruddh Khera, and Amir Husain, *Bayesian Neural Networks*, 2018, arXiv: 1801.07710 [cs.LG].

[17] Sumit Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, Towards data science, Dec. 2018, URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[18] Ashish, *Understanding Edge Detection (Sobel Operator)*, Sept. 2018, URL: https://medium.datadriveninvestor.com/understanding-edge-detection-sobel-operator-2aada303b900.

[19] Gary B, *Convolutional neural network*, DataScientest, June 2020, URL: https://datascientest.com/convolutional-neural-network.

[20] *Why is ReLu better than tanh and sigmoid function in artificial neural networks?*, ProgrammerSought, 2018, URL: https://programmersought.com/article/36091276265/.

[21] Dewang Nautiyal and Anuj Kumar, *Underfitting and Overfitting in Machine Learning*, geeksforgeeks, May 2020, URL: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/.

[22] Rinat Maksutov, *Deep study of a not very deep neural network. Part 5: Dropout and Noise*, Rinat Maksutov, July 2018, URL: https://rinat-maksutov.medium.com/deep-study-of-a-not-very-deep-neural-network-part-5-dropout-and-noise-29d980ece933.

[23] Chris McIntosh et al., "Fully automated treatment planning for head and neck radiotherapy using a voxel-based dose prediction and dose mimicking method", in: *Physics in Medicine Biology* 62.15 (July 2017), pp. 5926–5944, ISSN: 1361-6560, DOI: 10.1088/1361-6560/aa71f8, URL: http://dx.doi.org/10.1088/1361-6560/aa71f8.

[24] RayStation, *Machine learning automated treatment planning*, RaySearchLaboratories, Mar. 2019, URL: https://www.analyticsvidhya.com/blog/2020/09/why-gpus-are-more-suited-for-deep-learning/.

[25] Michal Drozdzal et al., *The Importance of Skip Connections in Biomedical Image Segmentation*, 2016, arXiv: 1608.04117 [cs.CV].

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell, *Fully Convolutional Networks for Semantic Segmentation*, 2015, arXiv: 1411.4038 [cs.CV].

[27] Dan Nguyen et al., *A feasibility study for predicting optimal radiation therapy dose distributions of prostate cancer patients from patient anatomy using deep learning*, 2018, arXiv: 1709.09233 [physics.med-ph].

[28] Dan Nguyen et al., "3D radiotherapy dose prediction on head and neck cancer patients with a hierarchically densely connected U-net deep learning architecture", in: *Physics in Medicine Biology* 64.6 (Mar. 2019), p. 065020, ISSN: 1361-6560, DOI: 10.1088/1361-6560/ab039b, URL: http://dx.doi.org/10.1088/1361-6560/ab039b.

[29] Ana M. Barragan-Montero et al., *Three-Dimensional Dose Prediction for Lung IMRT Patients with Deep Neural Networks: Robust Learning from Heterogeneous Beam Configurations*, 2019, arXiv: 1812.06934 [physics.med-ph].

[30] Liu Z et al., *deep learning method for prediction of three-dimensional dose distribution of helical tomotherapy*, Med Phys. 2019 May;46(5):1972-1983. doi: 10.1002/mp.13490. Epub 2019 Mar 30. PMID: 30870586.

[31] Kearney V et al., *DoseNet: a volumetric dose prediction algorithm using 3D fully-convolutional neural networks*, Phys Med Biol. 2018 Dec 4;63(23):235022. doi: 10.1088/1361-6560/aaef74. PMID: 30511663.

[32] Eyke Hüllermeier and Willem Waegeman, *Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods*, 2020, arXiv: 1910.09457 [cs.LG].

[33] João Caldeira and Brian Nord, "Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms", in: *Machine Learning: Science and Technology* 2.1 (Dec. 2020), p. 015002, ISSN: 2632-2153, DOI: 10.1088/2632-2153/aba6f3, URL: http://dx.doi.org/10.1088/2632-2153/aba6f3.

[34] Jyotirmay Senapati et al., "Bayesian Neural Networks for Uncertainty Estimation of Imaging Biomarkers", in: *Lecture Notes in Computer Science* (2020), pp. 270–280, ISSN: 1611-3349, DOI: 10.1007/978-3-030-59861-7_28, URL: http://dx.doi.org/10.1007/978-3-030-59861-7_28.

[35] Michael Morris et al., *Estimating the Uncertainty of Neural Network Forecasts for Influenza Prevalence Using Web Search Activity*, 2021, arXiv: 2105.12433 [cs.LG].

[36] Dan Nguyen et al., *A comparison of Monte Carlo dropout and bootstrap aggregation on the performance and uncertainty estimation in radiation therapy dose prediction with deep learning neural networks*, 2021, arXiv: 2011.00388 [physics.med-ph].

[37] Donggeun Yoo and In So Kweon, *Learning Loss for Active Learning*, 2019, arXiv: 1905.03677 [cs.CV].

[38] Mayank Agarwal, *Batch Normalization, Instance Normalization, Layer Normalization: Structural Nuances*, Aug. 2020, URL: https://becominghuman.ai/all-about-normalization-6ea79e70894b.

[39] Aaron Babier et al., *OpenKBP: The open-access knowledge-based planning grand challenge*, 2021, arXiv: 2011.14076 [physics.med-ph].

[40] Aaron Babier et al., *Knowledge-based automated planning with three-dimensional generative adversarial networks*, 2018, arXiv: 1812.09309 [physics.med-ph].

[41] Sebastian Ruder, *An overview of gradient descent optimization algorithms*, Sebastian Ruder, Jan. 2016, URL: https://ruder.io/optimizing-gradient-descent/.

[42] *tf.keras.optimizers.Adam*, Tensorflow, 2021, URL: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam.

[43] Prathmesh Patil, *Why GPUs are more suited for Deep Learning?*, Analytics Vidhya, Sept. 2020, URL: https://www.analyticsvidhya.com/blog/2020/09/why-gpus-are-more-suited-for-deep-learning/.

[44] Margerie Huet - - Dastarac, *Dose prediction for protontherapy using neural networks*, 2019, Prom. : Lee, John Aldo ; Sterpin, Edmond. http:// hdl.handle.net/2078.1/thesis:19554.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**