UCLouvain

École polytechnique de Louvain

Image modality conversion from helical CT to cone beam CT with deep learning model for adaptive radiation therapy

Author: **Olga Loshchakova**
Supervisors: **Ana Maria Barragán Montero, John Lee**
Readers: **Elena Lucia Borderias Villarroel, Christopher De Vleeschouwe**
Academic year 2020–2021
Master [120] in Data Science: Information technology

Abstract

Radiation therapy is a common technique to treat cancer patients. The aim of each radiotherapy treatment is to deliver a sufficient radiation dose to the tumour volume in order to kill cancer cells and at the same time not hurt the healthy tissues. Radiotherapy treatments usually last from few days to several weeks. At first, to plan treatment, a computed tomography (CT) of the patient is acquired, and all the treatment is based and planned according to this initial CT, and it is assumed that the internal organs do not change position and shape throughout the whole treatment. However, during treatment, not only the location, shape, and size of the tumour itself can change but also the location of other internal organs can change, making the treatment less effective. Thus, for an effective treatment, it is necessary to know the exact location at the beginning of the first exposure and before each subsequent session during the entire course of radiation therapy.

Adaptive radiotherapy takes into account all the above disadvantages of conventional radiotherapy. It is a well-known technique that acquires daily images of the patient and adapts the plan to the current anatomy, ensuring that the tumour is well covered and the organs are well-spared at any time of the treatment. The use of cone-beam computed tomography (CBCT) can reduce inaccuracy in dose calculation to the tumour and surrounding organs at risk, and further constantly adjust the dose and improve image-guided radiation therapy. The advantage of CBCT images compared to conventional computed tomography (CT) is that it takes less exposure to the patient's internal organs. Although CBCT allows the production of 3D images daily, the images suffer from severe artefacts and low contrast in soft tissue, resulting in inaccurate Hounsfield units (HU) values. These limitations make the CBCT-based adaptive planning process impractical. Therefore, it is very important to improve the image quality of CBCT.

Recently, deep learning has made great strides in image-to-image conversion tasks. However, it is generally difficult to obtain paired CT and CBCT images with exact matching anatomy for supervised learning.

This Master's thesis deals with a problem of producing high-quality synthesized computed tomography (sCT) images from low-quality CBCT images for prostate patients based on implementing an unsupervised deep learning method. For this purpose, a cycle-consistency generative adversarial neural network (CycleGAN) is used. The implemented CycleGAN learns the mapping between CBCT images and planning sCT images, preserving the anatomical structure from the original CBCT images and suppressing artefacts.

Acknowledgements

I am sincerely grateful to the brilliant team that has guided and supervised me all the time I was working on my thesis. I am very grateful for this experience, which has been changing me all this time and enriched me with new qualities and skills. To be honest, it was very difficult, but at the same time very informative, interesting and exciting. Thank you for your support and encouragement with all the kind words that inspired me to go forward.

I would like to express my special gratitude to Professor Lee. First, because he recommended to choose this topic. And thanks to him and the given opportunity, I gained colossal knowledge and tremendous experience in the field of medical imaging, radiation therapy, and, what is far more important, in working with deep neural networks. I really enjoyed working on this topic and received encouraging and positive support from him.

I will always be grateful to Ana for her supervision and support in any situations when something was going wrong or did not work out. And I am very thankful for the constructive comments on the presentations and the manuscript. Without them, I would not have been able to achieve these results, to get where I am, create this thesis.

My sincere thanks to Elena for her advice and help in getting my results, especially for the data that she processed for me. Also for the help with editing the manuscript and advice regarding different parts of the thesis.

Many thanks to Jean for the code that that he kindly shared with me, and which helped me in developing my own code. Thanks to Kevin for technical support and help with the subtleties of working with the server. Thanks to Romain and Cyprien for the deeply-professional expert assessment of my work, which later helped me to find error, inaccuracies in my work and, as a result, debug the programs I ran.

Thanks to the reader, Professor Christopher De Vleeschouwe, for agreeing to be a member of the commission of my thesis.

Most of all, many thanks to my big family. To my husband, Alexander, who always supports me, helps me in difficult times and always inspires me to study and grow. For your ideas during these several years of study and especially throughout the thesis work. Thanks to my mother-in-law, Eugenia, for all the help and moral support I received from her.

List of Abbreviations

ART	Adaptive radiation therapy
IGRT	Image-Guided Radiotherapy
CT	Computed Tomography
CBCT	Cone-Beam Computed Tomography
sCT	Synthesized Computed Tomography
HU	Hounsfield Units
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NN	Neural Network
ANN	Artificial Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
GAN	Generative Adversarial Neural Network
CycleGAN	Cycle-Consistency Generative Adversarial Neural Network
MSE	Mean Squared Error
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
PNSR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure

Contents

Chapter 1

Introduction

According to the most recent statistics on population-based cancer occurrence, the number of new cancer cases in 2020 is 19.3 million in the world. That is, over the past two years, the number of new cases has increased by 6.6%[1][2]. In particular, prostate cancer is the second most common cancer diagnosis after lung cancer in men (in 2020, 1,414,259 new cases worldwide) and the seventh leading cause of cancer deaths worldwide (in 2020, this is 3.8% of all cancer deaths in men).[2]

In oncological practice, radiation therapy is one of the most widely employed therapies for the treatment of cancer, which can be used as a stand-alone treatment method. However, more often, it is combined with other methods such as surgery, chemotherapy and immunotherapy. The basic principle of external radiation therapy is the use of ionizing radiation to kill cancerous cells. The delivered radiation dose is measured in Gray, which is the absorption of one Joule of ionizing radiation per one kg of mass.

The radiation dose has a detrimental effect on cancer cells and harms the surrounding normal tissues. Therefore, the goal of every radiation therapy treatment is to deliver a high dose to the tumor sufficient to destroy it and at the same time reducing the dose to normal organs and tissues while maintaining their viability.

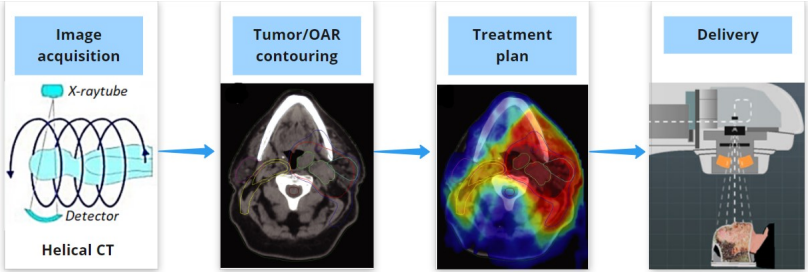


Figure 1.1: Radiotherapy workflow [3][4]

In practice, maximizing the probability of tumour control (TCP) and minimizing the probability of complications in normal tissue (NTCP) must be balanced. This balance leads to optimization treatment plans and is used to calculate the radiation dose and the number of fractions in the course of treatment.

As shown in Figure 1.1, the conventional radiotherapy workflow consists of several consecutive steps and is carried out by a team of specialists. The first step is to acquire imaging of the patient (planning CT image) before treatment, usually using a helical computed tomography (CT) to pinpoint the tumor location. This initial CT scan is used to construct the treatment plan. The radiation oncologist then delineates the tumor boundaries in 3D geometry. After delineating the tumor and the organs at risk of toxicity, the dose is calculated, and a treatment plan is drawn up. The total radiation dose is divided into so-called fractions delivered over a certain number of days (usually one fraction per day). It can take from a few fractions (5-10) to longer treatments (30 fractions). After the approval of the treatment plan, irradiation is performed under the clinical supervision of a physician.

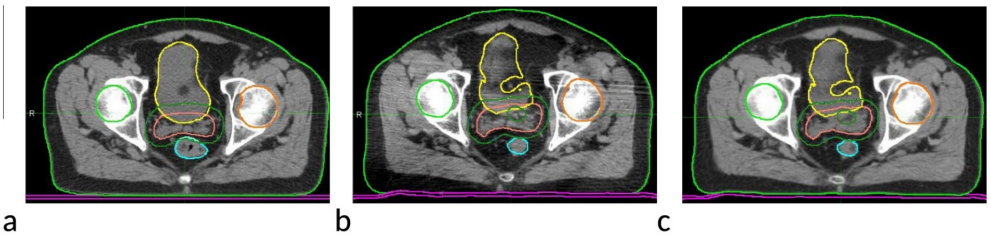


Figure 1.2: Anatomical changes during the treatment. [7]

The drawback of this treatment method is that it assumes that the patient's anatomy is invariant during treatment. However, in practice, the anatomy can change between the planning CT and delivery and throughout radiation therapy. These changes can be in the order of centimetres. The anatomical changes may be due to, for example, a change in the shape and size of the tumor, or patient's weight loss or gain, which can affect the displacement of organs and the tumor. [9] In addition, because of the location of the prostate, the physiological filling of the bladder causes random fluctuations in the position and shape of the prostate, making proper setup and treatment challenging (Fig.1.2).

The adaptive radiation therapy (ART) is a well-known technique that acquires daily in-room imaging of the patient and adapts the plan to the current anatomy, ensuring that the tumor is well covered and the organs are well-spared at any time of the treatment. [10]

The ART workflow is a closed-loop radiation therapy process and consists of adding the following three technical parts to the conventional radiation therapy

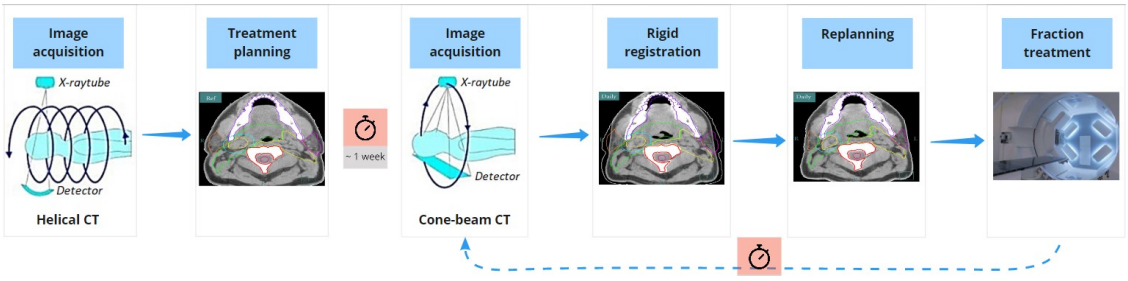


Figure 1.3: Adaptive radiotherapy process

workflow: daily image guidance, dose validation, and plan adaptation (Fig. **??**) [8]. A daily cone-beam CT (CBCT) scan, obtained prior to the fractional treatment, is used for daily contouring of the tumor and organs at risk. The CBCT image is registered with the planning CT to estimate the daily dose delivered, and an adapted treatment plan is drawn up.

CBCT uses a cone-beam of X-rays and has several benefits concerning a traditional CT scanner, including the ability to acquire up-to-date volume information about patients more rapidly and with less exposure to patient's healthy organs.

The complexity of adaptive radiotherapy is resource and labour-intensive processes (re-contouring, image registration and re-planning) that place greater reliance on existing human resources. At the same time, most often, CBCT images have poor quality and lower soft-tissue contrast than CT, resulting in inaccurate Hounsfield units (HU) values. Also, CBCT suffers from severe artefacts, some inherited from CT, others being specific: extinction artefacts, beam hardening artefacts, partial volume effect and exponential edge-gradient effect, aliasing artefacts, ring artefacts and motion artefacts (misalignment artefacts), noise, and scatter. In wide-angle cone beams, a geometric artefact distorts both ends of the cylindrical field of view.

Over the last five years, deep learning have made significant progress in developing AI applications in medical imaging, aiming to automate different clinical practice steps or provide assistance with clinical decisions[12][13]. Deep neural networks (DNNs) have been used to transform different image modalities, which included correcting artefacts in CBCT images.

In this project, we implement an unsupervised deep learning method to create high-quality synthetic computed tomography (sCT) images from low-quality CBCT images for prostate patients. Good results in CBCT-to-CT conversion could enhance further development of adaptive radiotherapy and to make the work of specialists more effective.

We start with a network that is optimized on image-to-image translation (of zebra-to-horse type), and our goal is to check whether it can be adopted to CBCT-to-CT conversions. We consider several models, including the modified U-Net

model, and make quantitative assessment of their performance on CBCT-to-CT conversion.

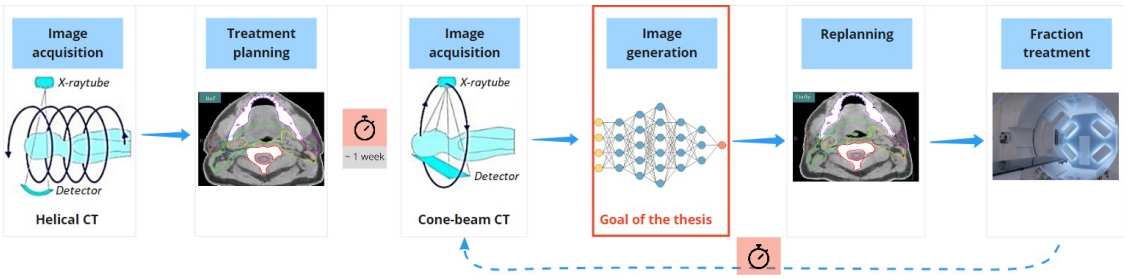


Figure 1.4: Adaptive radiotherapy process with deep learning.

The structure of this manuscript is organized as follows. Chapter 2 introduces machine learning, deep learning, and applications of deep neural networks in medical imaging. Chapter 3 covers literature reviews of previous works that have been published. Chapter 4 describes the methods and materials proposed in this thesis, followed by the results in Chapter 5. Finally, a discussion and a conclusion close the document in Chapter 6.

Chapter 2

Context

2.1 Radiation therapy and adaptive workflow

2.1.1 CT acquisition

General process

Computed tomography (CT) is a modern diagnostic technique based on the use of X-rays that allows obtaining images of human body organs [16]. The working principle is based on the fact that different human body tissues absorb X-rays to different extents.

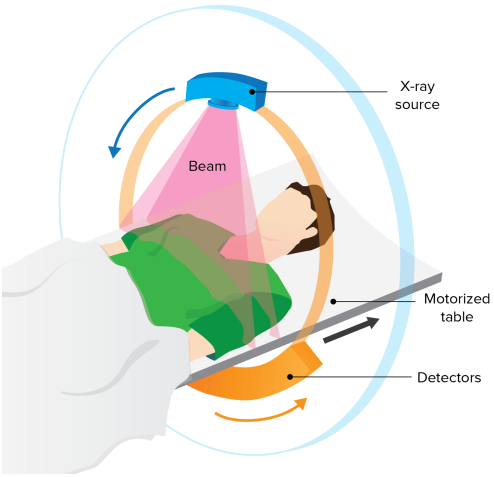


Figure 2.1: Computed tomography scanner ingredients are an X-ray tube, detectors and a motorized table. [15].

There are several popular types of CT scanners (sequential, spiral, etc.), all

of them share the main principles of work. Inside the CT scanner there is a rotating X-ray tube (an X-ray source on Fig.2.1) that emits multiple narrow beams that pass through the patient's body. On the opposite side, there is a sensor (a detector on Fig.2.1) that absorbs the transmitted beams. Since body tissues of different densities absorb X-rays to different degrees, the transmitted X-rays differ in intensity at the output. The patient is placed on a special motorized table, which is then driven through a large scanner. As the table is moving along the scanner, the device takes a sequence of scans.

In the case of sequential CT scanners, the scanner rotates around the body, and the signals are transmitted to the computer, where specialized medical software compares the intensity of the emitted and absorbed X-ray beams. These measurements are used in a reconstruction algorithm to obtain cross-sectional images (2D slices). The scanner moves along the body and obtains different cross-sections, which are then stacked and provide a 3D image of the area of the patient's body under study. CT images can be viewed from the feet to the head (axial plane), from the side (sagittal plane), and frontally (coronal plane) (see Fig.2.2).

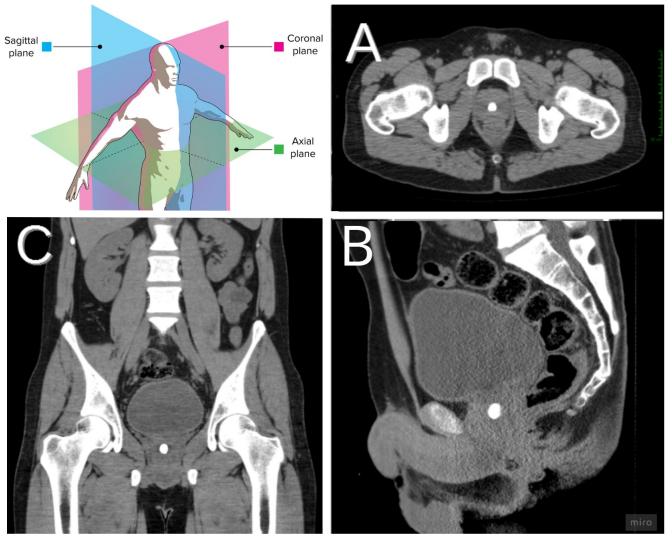


Figure 2.2: CT abdomen and pelvis: A) axial plane, B) sagittal plane, C) coronal plane [15] [20].

The advantage of CT techniques is that it allows obtaining a detailed scan of the human body. However, the disadvantage is that the dose needed to obtain the image is relatively high. In addition, according to some research, the procedure might change or damage the DNA of the patients, resulting in radiation-induced cancer [17].

Contrast agent

Sometimes CT can be done with contrast agents. For example, if it is necessary to obtain a 3D image of the prostate gland, the patient is injected intravenously with a contrast agent containing iodine (Fig.2.3). The contrast on the tomogram allows for a high-quality oncopic search and makes the position and contours of the masses more visible.

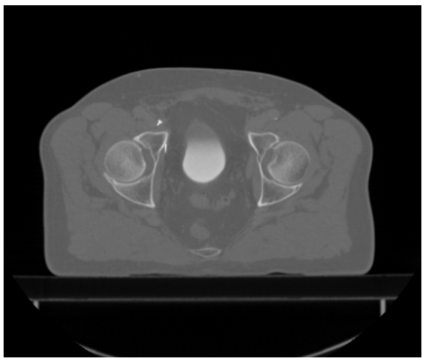


Figure 2.3: Contract agent in the bladder

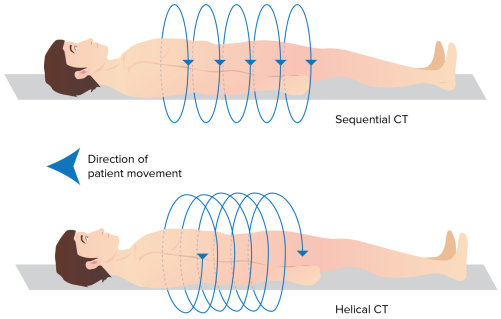


Figure 2.4: Helical and sequential (conventional) CT [15].

Spiral computed tomography

Another popular type of computed tomography is a spiral computed tomography (spiral CT, or helical CT). During this procedure two actions are performed simultaneously: continuous rotation of the X-ray tube around the patient's body and the translational movement of the table with the patient inside the tomograph [16] [18]. When the X-ray tube moves, its trajectory takes the form of a spiral. This technology made it possible to significantly reduce the time of the study and thereby reduce the radiation load on the patient's body (see Figure 2.4).

Hounsfield units

To evaluate quantitatively a density of tissues one uses the so-called Hounsfield units. A three-dimensional CT image consists of voxels (parallelepipeds), each with a certain degree of radiation absorption. The absorption rate depends on the density of the soft tissue. The more radiation is absorbed, the brighter the object (bone tissue) appears on tomograms; the less the structures absorb radiation, the darker the image (liquid, air). Multiple voxel projections taken from different angles allow density to be calculated based on measured radiation attenuation. Then, each voxel is averaged over its thickness and projected onto the screen as a 2D

pixel [19]. For a quantitative, visual assessment of the density of body structures, the Hounsfield scale, named for the inventor of computed tomography, Godfrey Hounsfield, is used. Scale units are defined as Hounsfield units (HU). This scale is a scale of X-ray attenuation relative to water (0 HU) ranging from -1.000 HU (air) to +3.000 HU (dense bone)[16]. For instance, positive values correspond to soft tissue (+40 to +80 HU), cortical bone (+700 HU), cancellous bone (+3,000 HU). Negative values on the Hounsfield scale correspond to lung (–500 H), adipose tissue (–90 to –120 H), air (-1000 HU). The HU value for a substance x can be calculated using the formula:

$$HU_x = \frac{\mu_x - \mu_{water}}{\mu_{water} - \mu_{air}} * 1000,$$

where μ_x is the linear attenuation coefficient of the substance x, and μ_{air}, μ_{water} - for air and water at standard conditions.

Artefacts from implants

Computed tomography can be performed as well with patients who have metal implants. Attenuation of implants depends on the metal: titanium has a CT value of +1000 HU, whereas cast iron steel can fully suppress X-rays and, hence, is the cause of line artefacts on tomograms (Fig.2.5). Artefacts arise from abrupt transitions between low and high density materials [16].

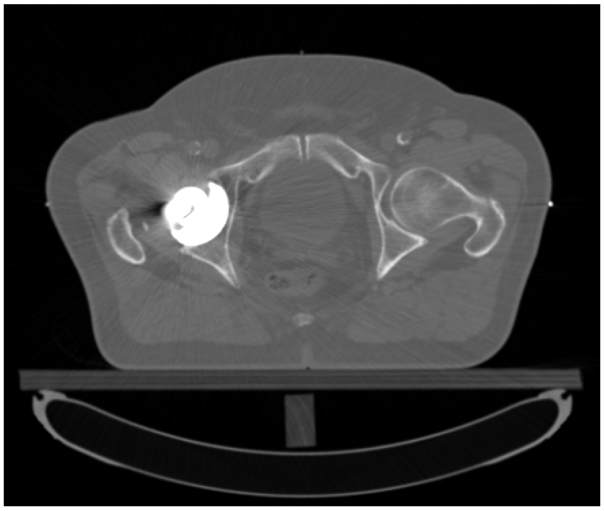


Figure 2.5: Line artifacts from the implant on the CT image.

2.1.2 Cone-beam CT

Basic technique

The cone-beam computed tomography (CBCT) principle is the same as that of more conventional CT besides differences in the form of the beam radiated from an x-ray tube, the shape of the detectors, and reconstruction techniques [23] (Fig.2.6). The beam can cover an extended volume with just one pass because of its cone form. Then a computer processes the transferred x-ray images and converts them to 3D images. The values of a CBCT voxels are saved in HU, just like CT scans.

In terms of the amount of radiation that patients are exposed to, there is a significant distinction between the two scanners. A CT scan employing a CBCT scanner can be done with reduced radiation doses. Panoramic radiography is an example of this. A conventional CT scan involves separating the upper and lower jaw scans that irradiate the patient 200-300 times higher than acceptable for this X-ray. The exposure is doubled if both jaws are scanned. CBCT scans both jaws at once, decreasing exposure.

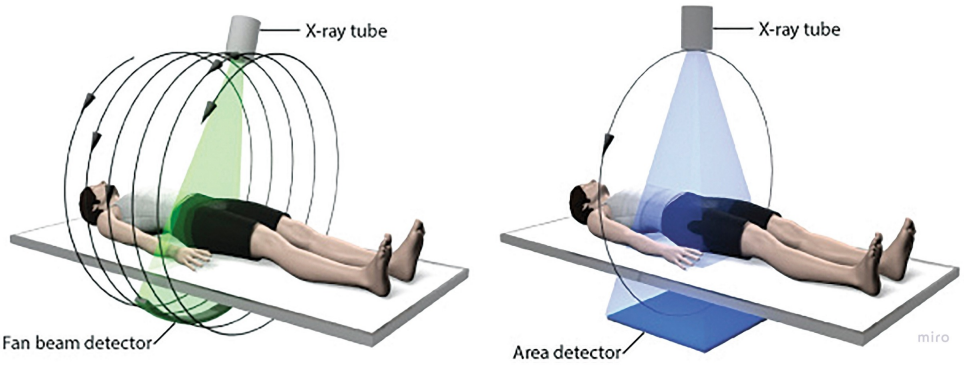


Figure 2.6: Left: Helical CT imaging. A fan beam of X-rays and several spins are used to image a thin slice of the patient. Right: Cone-beam CT imaging. A detector with a large area produces an extended volume of the patient's body in a single pass. [22]

However, CBCT has several disadvantages. Because of the low contrast resolution, some softer tissues cannot be seen. Artefacts such as noise, beam hardening, and scattering are also more prone to occur [24]. Scattering, mainly, is a significant restriction that may preclude the use of CBCT for radiotherapy.

2.1.3 Registration of images

By image registration, we understand a meaningful spatial mapping of one image into another one. Image registration finds its application in several areas, such as

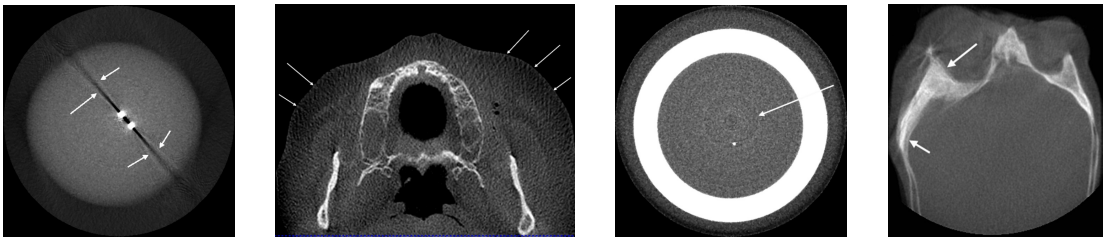


Figure 2.7: 1st: Exponential edge-gradient effect (EEGE); 2nd: Typical aliasing patterns (Moire patterns); 3rd: Ring artefacts; 4th: Typical double contours (arrows) induced by patient movement

merging medical images of different modalities or improving dis-aligned or distorted images [27]. The registration procedure aims to discover the transformation function that maps a moving image/volume onto a fixed image/volume in such a way that the difference between the target image and the deformed image is minimized. [26]. Using image registration, we can compute a transformation between the initial CT scan of the initial treatment planning and the new CBCT scan; in this case, the moving image is the initial CT image, and the fixed one corresponds to the CBCT image.

There are different approaches to registration, which depend on the type of transformation, the two main of which are rigid and non-rigid registration [25].

A rigid registration matches two images with a rigid transformation, i.e. a translation, a rotation or a combination of both. It means that the proportions and shapes of the objects are preserved. Non-rigid or elastic registration stretches one image volume to fit another image volume (Fig.2.8).

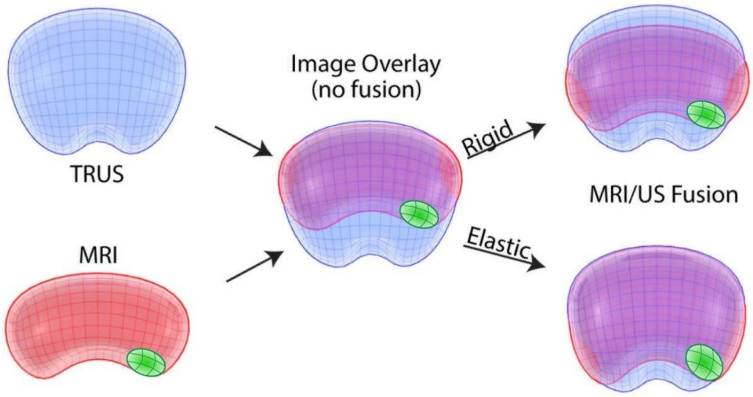


Figure 2.8: The example of rigid and elastic (non-rigid) registration of magnetic resonance images (MRI) and transrectal ultrasonography images (TRUS) for prostate biopsies. Credit: `https://pubmed.ncbi.nlm.nih.gov/24298917/`

2.1.4 Delineation of contours

The next step is delineating the contours of the tumor and the organs at risk (OAR). It can be done in three different modes: manually (using specific software), semiautomatically (when a user have some control over the delineation process) and fully automated (computer algorithms completely do process); these methods differ in the amount of time needed to spent to process an image.

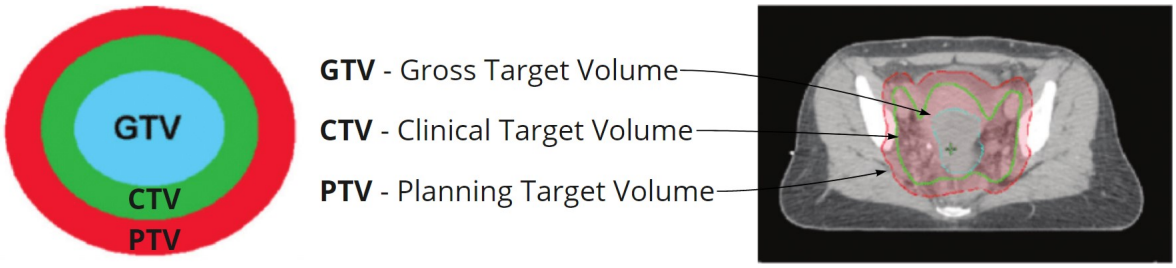


Figure 2.9: Delineation contours principle in radiotherapy planning. Left: diagram of tumour region separation on sub-regions, Right: axial CT image where a) the gross tumor volume GTV (blue), b) the clinical target volume CTV (green), and c) the planning target volume PTV (red) are defined. [28]

As illustrated in Figure 2.9, the target tumor volume is divided into three different volumes, each irradiated with the relevant intensity:

a) the visible macroscopic tumor, or the gross tumor volume (GTV),

b) the clinical target volume (CTV) – the GTV volume + invisible microscopic tumor spread, and

c) the planning target volume (PTV) – the CTV volume + volume to take into account geometric errors and other factors such as a tumour or organs motion and patient setup inaccuracy in the treatment room.

2.1.5 Optimization of the treatment and dose calculation

Once the tumour and OAR are localized, maximum and minimum dose restrictions are imposed on the regions of interest (an upper bound for the radiation dose that will be unavoidably delivered to the healthy cells and a lower bound for the radiation dose that must be exposed on the tumour cells). As a result, an optimization problem arises, with the objective function often being the sum of individual dose objectives for each region of interest. A dosimetrist establishes the optimal set of parameters to optimize a dosage distribution in accordance with clinical requirements. This parameter tuning process is iterative, involving the usage of specialized treatment planning software (e.g. RayStation from RaySearch Laboratories) and the participation of a dosimetrist, who manually inserts the dose

objectives and their relative importance weights, taking into account the organs at risk and PTV. Radiation tolerance varies by organ and is determined by tissue characteristics. This information is also used to prescribe a dose of radiation that will not cause them any harm. After the software gives recommendations for the dose distribution, the dosimetrist analyzes it according to the patients' specific anatomy and, if needed, tunes the parameters of the objective function, and the process repeats until a satisfactory dose distribution is obtained.

While calculating the radiation dose, it is important to take into account tissue inhomogeneities, the imaging quality and calibration of Hounsfield Units. In adaptive radiotherapy, one uses the daily CBCT imaging, which might lead to incorrect dose calculation due to inaccurate HU values of CBCT images. The thesis aims to generate high-quality sCT images from low-quality CBCT images, with corrected HU values for precise radiation dose calculation.

2.1.6 Evaluation of the treatment

After an optimal solution has been established, the physicians evaluate the dose distribution. The treatment plan with the dose calculated will be reviewed and approved again. There are different metrics to compare the prediction to the accurate dose distribution. To summarize the dose distribution to the target volume and OARs, dose volume histograms (DVH) are frequently employed (Fig.2.10).

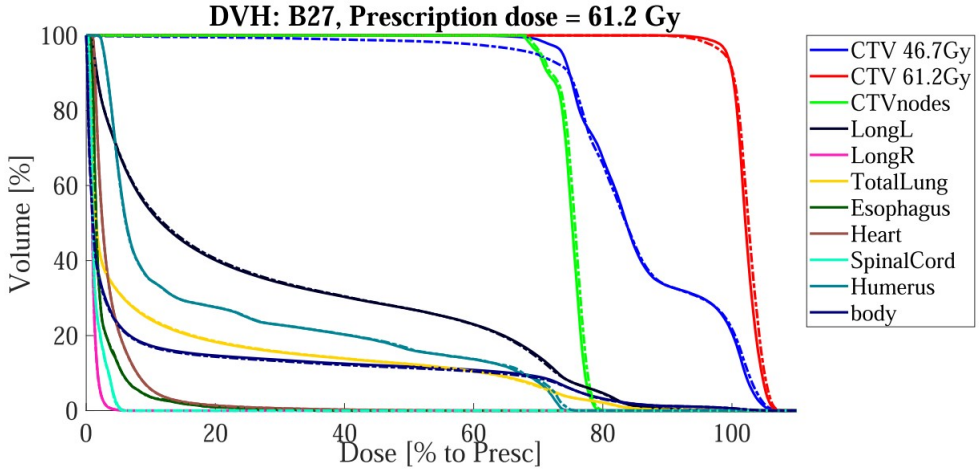


Figure 2.10: An example of the dose-volume histogram (DVH) is illustrated for target volume and OARs of synthetic CT (solid lines) and rescan CT (dashed lines). The curves mean the dose (%) received by the volume (%) of the interest) (for each dose on the coronal axis).)[79].

2.2 Machine learning tools for medical imaging

The fact that artificial intelligence (AI) technologies are today the number one topic in the IT industry can be judged not only by enthusiastic publications in the media, scientific journals and numerous projects in this area but also by the scope of AI penetration into almost all areas of modern life - from medicine, expert systems and scientific research to industrial robotics and self-driving vehicles. In recent decades, machine learning and neural networks have been actively developing and improving in the field of image analysis and processing. The medical community shows its great interest in artificial intelligence. Alongside radiology, which is undoubtedly the dominant field, medical imaging analysis has found applications in many areas of medicine, including digital pathology, ophthalmology, dermatology and endoscopy, which is represented by many practical applications of artificial intelligence in medical imaging. The most common medical image analysis tasks where AI systems are implemented are segmentation and classification, which help doctors with diagnosis, namely, classification is a supervised machine learning task, segmentation is a pixel-wise machine learning task, and from the medical perspective, diagnosis often relies on classification.

The purpose of this chapter is to introduce the basic concepts of artificial intelligence, as well as the main concepts and modern methods of machine learning and their application in medical imaging.

2.2.1 Machine learning

Machine learning (ML) is a part of the artificial intelligence (AI), which is defined as an intelligence demonstrated by machines, as opposed to natural intelligence. The specificity of ML is that it is able to learn from experience and from the use of data. This is achieved by creating a model that depends on various tunable parameters; the process of tuning the parameters in such a way that the model works well on different pre-loaded examples is called the “training” of the model. After several iterations of training and tuning, the final model is evaluated on a test set, used to simulate how the model will perform when faced with new, unseen data.

The main goal of the models are to be able to generalize their learned expertise, and deliver correct predictions for new, unseen data. When training on a training set, a model might run into overfitting – photographical memorization of the seen dataset, while performing poorly on yet unseen, but obviously related cases. That is why for training an adequate model it is important to include a separate training set, on which the model’s generalization ability is estimated. The feedback from the validation set is used as a feedback for further tuning of the model. We can say that machine learning, i.e., the process by which computing systems achieve

the capability to perform sophisticated tasks, essentially consists of learning by examples, much as we human beings do.

Note that learning by examples requires both data and computational power to be successful, and was not possible in the beginning of the computer era, when the first early attempts towards AI systems were based on the definition and application of rules and were often unsatisfactory because rules typically have numerous exceptions, which are hard to implement and even to formalize.

2.2.2 Major branches of machine learning

At a basic level, classical Machine Learning can be classified according to the type of supervision into four broad categories: supervised, semi-supervised, unsupervised and reinforcement learning (Fig.2.11).

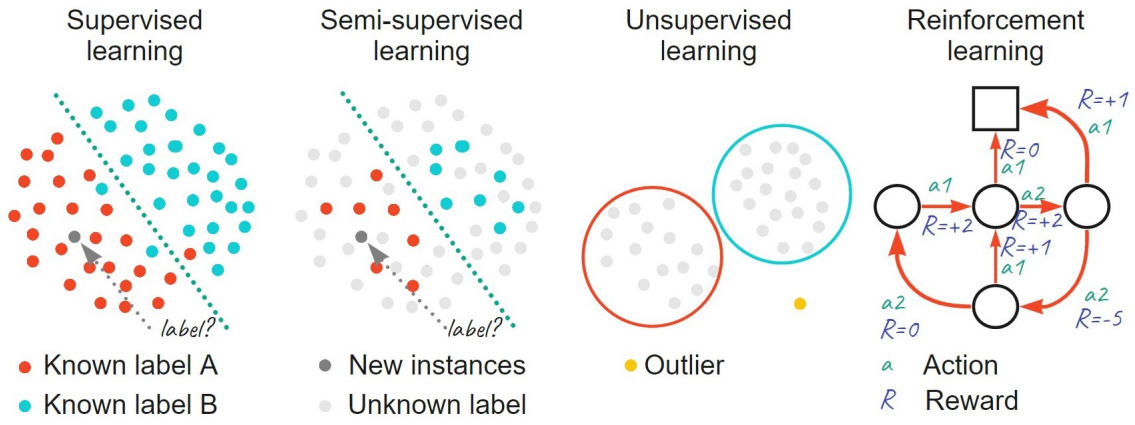


Figure 2.11: Machine learning types (inspired by [13]).

Supervised Learning. In a supervised learning scenario, the process consists of learning how to map features of the input to known targets (also called labels or annotations) based on input-output examples that are most often annotated by humans.

Based on the rules discovered in the labelled training dataset the trained model is thus used to produce correct labels on new previously unseen datasets [29].

The supervised approach is used in specific data processing tasks, such as in classification problems (with two or more classes), regression, and in more exotic variants, particularly, sequence generation [30], object detection [31] and image segmentation [32], [33]. For example, in the problem of classifying skin lesions in accordance with malignant neoplasms, a classification model is used and the problem of identifying risk factors for cardiovascular diseases in photographs of the fundus of the retina is divided into several subtasks, with respect to the data type:

predicting the binary risk factors (a classification model) and the continuous risk factors (a regression model) [34], [35], [13].

Unsupervised Learning. Unsupervised learning is the opposite of supervised learning, and means that the training data is unlabelled and only input data without corresponding outputs is available. Unsupervised learning is aimed at analysing the data and uncovering patterns in the data without the guidance.

This model is especially useful in the clustering task, when it is necessary to select separate groups of data or clusters, in the task of dimensionality reduction, which is aimed at simplifying data without losing the main information, and in the task of detecting anomalies or outliers.

Over the past few years, unsupervised learning has resulted in the development of useful medical imaging applications, such as domain adaptation [36], [37], data generation [38], [39] or even image segmentation [40].

Semi-supervised. There is a hybrid model between supervised learning and unsupervised learning called semi-supervised learning. Semi-supervised learning algorithms are useful when many of the target outputs are missing and only partially labelled data is available. If the cost of labelling all the remaining examples is too high, one turns to the semi-supervised setup.

In semi-supervised learning methods, as in unsupervised learning, data is divided into clusters based on the labels of the annotated data [41] (see Fig.2.11).

Clinical applications have been developed for generating or transferring images from one class to another under semi-supervised settings, as well as segmentation or classification of images with partly annotated data [42], [43].

Reinforcement Learning. In reinforcement learning, there is an agent who interacts with its environment and learns to choose the actions that will allow it to receive the highest reward. The environment either gives a reward for these actions and the agent continues to take them or penalizes it in the form of negative rewards.

The power of reinforcement learning algorithms can be seen from the fact that at different times they have won human world champions in such games as chess, Go, Atari, shogi, Dota 2 [44]. Note, however, that in chess, computers started to outperform humans even before an active implementation of deep neural networks (Kasparov vs IBM's deep blue 1996, 1997; the deep blue was based on alpha-beta search algorithm). Nevertheless, the current chess champion Google's alpha zero is implemented by using deep neural networks.

The first progressive reinforcement learning applications in the medical domain introduced a couple of years ago mimic physician behaviour for a number of everyday tasks, including treatment planning.

Other machine learning types. Along with these four classical basic types of learning, there is also a recent "self-supervised" learning, as well as other strategies

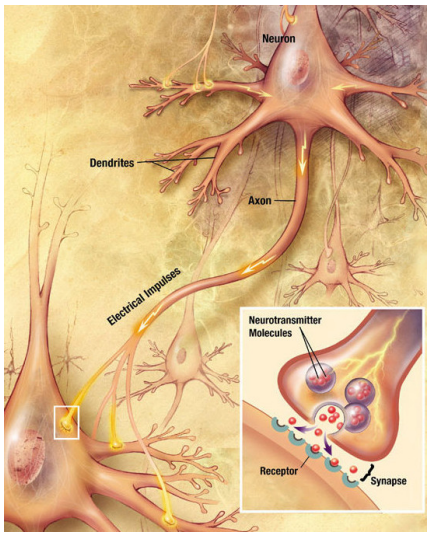
that allow us to reuse or combine previously trained models (transfer learning and ensemble learning) (see [13] for more details).

2.3 Neural networks

The development of AI systems began in 1943 when the neuropsychologist Warren McCulloch and the mathematician Walter Pitts put forward a new neural framework and proposed a mathematical model of an artificial neuron by analogy with a biological neuron (*A logical calculus of the ideas immanent in nervous activity*, Warren S. McCulloch and Walter Pitts, 1943). Almost ten years later, in 1956, at a conference at the University of Dartmouth, computer scientist John McCarthy coined the term "*artificial intelligence*". Based on the ideas of McCulloch and Pitts, as well as the work of neurophysiologist Donald Hebb on the basic principles of neuron learning, the Perceptron neural network, which simulates the process of human perception, was invented by neurophysiologist Frank Rosenblatt in 1957.

2.3.1 Artificial neuron

Artificial neural networks use a biological analogy to neurons that reside inside the brain. Biological neurons are very complex, but in a simplified form they can be represented as follows: a neuron has a nucleus in which an electrical charge is accumulated. The charge enters through the processes of the neuron, which are called dendrites; through these processes, signals from other neurons come. After a certain amount of charge has been accumulated inside the nucleus, the neuron fires and emits an electrical signal to the output branch, which is called the axon. The axon, in its turn, attaches to the input dendrites of other neurons.



Chemical synapse schema. Credit: https://en.wikipedia.org/wiki/Neuron

The riveting is done through the so-called synapses, which can change the transmitted signal. The signal can increase or decrease in the synapse.

An artificial neuron has several inputs of analogues of dendrites, to which input signals x are supplied (Fig.2.12). An artificial neuron has one output y through which an output signal is issued by analogy with an axon in a biological neuron. Each input is given a certain weight w, they are analogues of a synapse. A

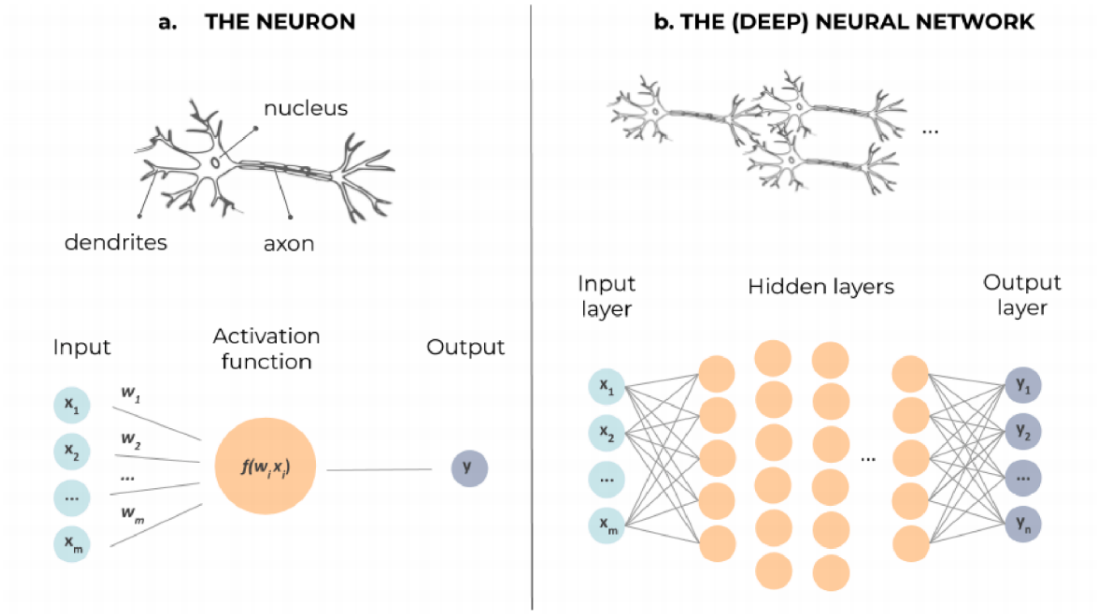


Figure 2.12: Biological neuron vs artificial neuron [13]

large positive weight amplifies the signal, similar to an excitatory synapse, while a negative weight attenuates the input signal, as does an inhibitory synapse. The neuron's output is calculated using the following formula:

$$a = f(\sum_{i=1}^{N} w_i x_i)$$

The input signal of the neuron arriving at each input is multiplied by the weight, then all these products are added and transmitted to the input of some nonlinear function, which is called the activation function. The activation function determines whether the neuron fires or not.

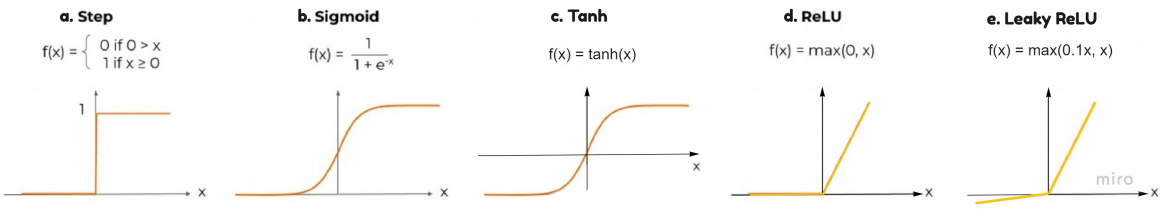


Figure 2.13: Activation functions (inspired by [13]).

In the McCulloch-Pitts model, the so-called Heaviside step function (Fig.2.13 (a)) was used as the activation function. If the input signal is less than a given

level, then the output of the neuron will be zero, and after the signal has reached a certain level, the neuron immediately outputs one. It is also possible to use other activation functions, often sigmoidal functions (b) are used, which perform not as abrupt switching as a step, but more smoothly. A logistic function as well as a hyperbolic tangent (c) or ReLU (d) (LeakyReLU (e)) can be used as an activation function (Fig.2.13).

2.3.2 Artificial neural networks (ANNs) and deep neural networks (DNNs)

Artificial neural networks are now one of the popular machine learning methods that are used in practice. A neural network is built from simple computational elements - artificial neurons (see Section 2.3.1.

McCulloch and Pitts in their work proposed to combine artificial neurons into neural networks, that is, the output signal of one neuron to transmit to the input of another neuron.

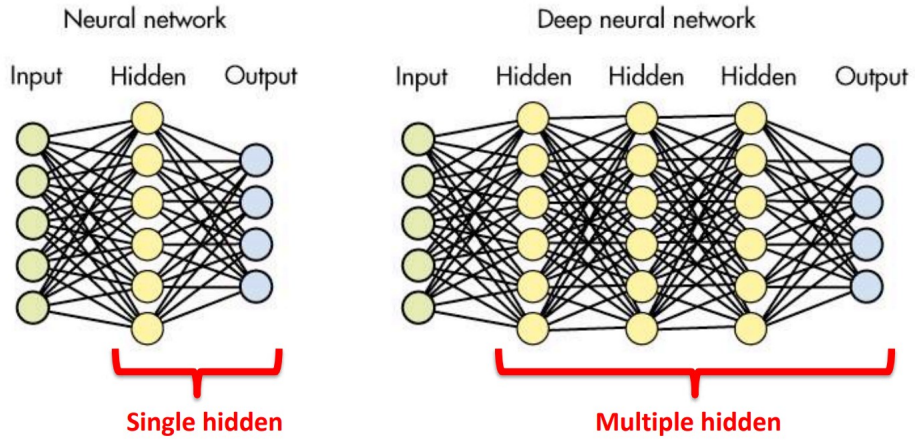


Figure 2.14: Shallow vs Deep neural networks (taken from the lecture on ML course (UCLouvain))

A network can consist of a large number of neurons combined into layers, where the input layer of neurons receives signals from the outside world, while the output layer produces signals to the outside world. And the internal layer of the neural network, which is called hidden because what happens in it is not visible from the outside (external environment), receives input signals from the neurons of the input layer and outputs signals to the output layer. Neural networks can include more than one hidden layer and such networks are called deep neural networks (Fig. 2.14).

In traditional machine learning we need to select some features in the available data (a.k.a. $\sim$ dimensions, or variables) that are useful to solve the problem at hand. Often, a person chooses or builds manually the important features for various tasks. And if someone manages to find such suitable handcrafted features, then the problem is likely to be solved more or less easily. The advantage of deep NNs is that they automatically separate the necessary data from the unnecessary ones and are able to choose the right features. On the other hand, since a neural network has to process much more data compared to other machine learning methods, the use of neural networks requires much more computing power.

Over the past few years, there has been a dramatic rise in the popularity of artificial neural networks in medicine. For example, Imperial College London uses neural networks to diagnose brain damage, and Microsoft is developing glasses for blind people to help them navigate their surroundings. These glasses have a built-in camera and headphones. The neural network analyzes the data from the camera and tells the person what to do. For example, if a blind person picks up a menu in a restaurant, then the neural network analyzing the data from the camera can read the menu to the person by voice into the headphones.

Why is this happening now, while most of the theory of neural networks was invented in the last century, starting from the 50s? There are two main reasons: the increased computational power of modern computers and a huge amount of accumulated data.

Nowadays, we see a significant increase in computer performance. Even in small devices, there are multi-core processors with high computational power and graphics accelerators, which can be used for computer games and computations, including training neural networks. Thus, if it took days or even months to train a network earlier, we can now train a network of the same complexity in minutes.

Secondly, training deep neural networks requires a huge amount of data. Thus a huge amount of data that have been stored in electronic databases within several past decays plays an important role in training DNNs. Indeed, unlike the conventional AI algorithms, where performance improves with the amount of available training data only up to a certain limit, after which the performance reaches the plateau and additional training data does not lead to a further improvement, the situation is cardinally different with neural networks. Thus, a sharp increase in the amount of accumulated data made it possible to train neural networks to solve complex problems (such as self-driving cars) and, as a side-effect, to deal effectively with big data problems.

Apart from these two main factors, another one is the availability of convenient ready-to-use deep learning software systems (Keras, TensorFlow, PyTorch, etc.), which simplifies the development of new systems.

2.3.3 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are created to mimic the human visual system and are often related to analyzing images and pattern recognition. They were invented to replace standard deep neural networks, which proved to be inefficient for such tasks.

Deep neural networks consisting of fully connected layers are hard to apply to large images since they require a large number of parameters (trainable weights), and the training processes become unfeasible. To solve this problem, scientists have proposed to deploy convolutional neural networks (CNNs).

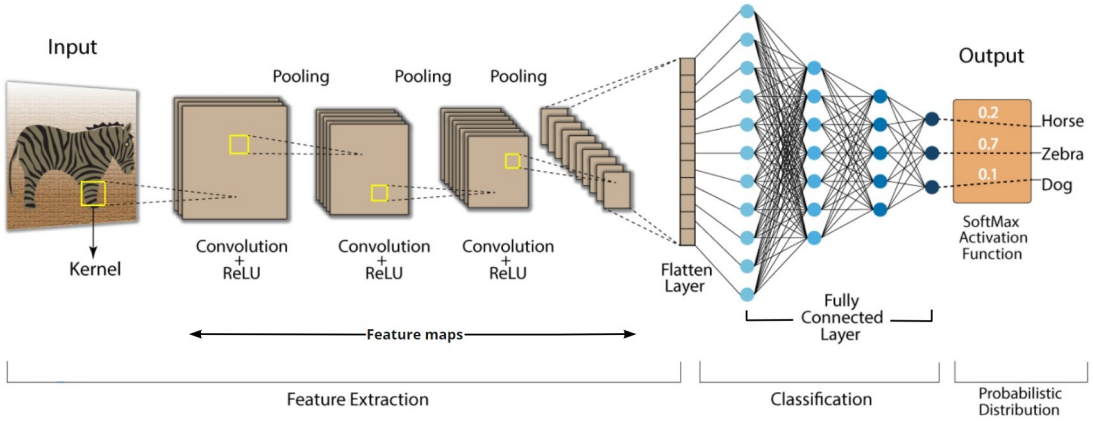


Figure 2.15: The architecture of Convolutional neural network

In contrast to the fully connected layers, which treat the pixels of the whole image at the same time, the convolutional layer splits the whole image into small pieces, and then treats them independently. This brings two important advantages. First, that allows to significantly reduce the overall number of trainable parameters and thus makes the training process feasible. Second, it makes convolutional neural networks to be translation invariant. Indeed, with a fully connected neural network, shifting a picture would require to retrain the whole model in order to recognize a new input. As opposed, the localized nature of the convolutional networks would recognize a shifted image without retraining. For example, if an object has been found in one part of the image, then its interpretation when it is located in another part of the image would be the same. To explain this property of CNNs, let us look at what different layers of CNNs do. Typically, shallow layers of CNNs (i.e. the ones closer to the input) capture low-level details and patterns, such as straight segments and edges; deeper layers (which are closer to the output) might capture more complicated patterns, such as animals shapes. Another advantage of CNN concerning fully connected neural networks is that CNN analyzes the image locally and identifies similar structures regardless of where in the image they are situated.

This explains a translation invariance property of CNNs.

Figure 2.15 shows a classic CNN architecture for an image classification problem that consists of basic layer types — an input layer, convolutional layers with activation function, max-pooling layers, fully connected layers, and an output layer. Now let us at these layers in more details.

Convolutional layer

Convolution is a technique for extracting relevant data from images. Convolution is performed by applying a filter to the input, then element-wise multiplication is performed, and the results added added together. The filter overlay is shifted one step to the right (or according to the setting of the stride), and the same calculations as above are performed to obtain the following result.

There are different options how one can treat the boundary of the image. The stride setting determines the number of cells by which the filter moves to the right and down in the input for the next calculation to be performed.

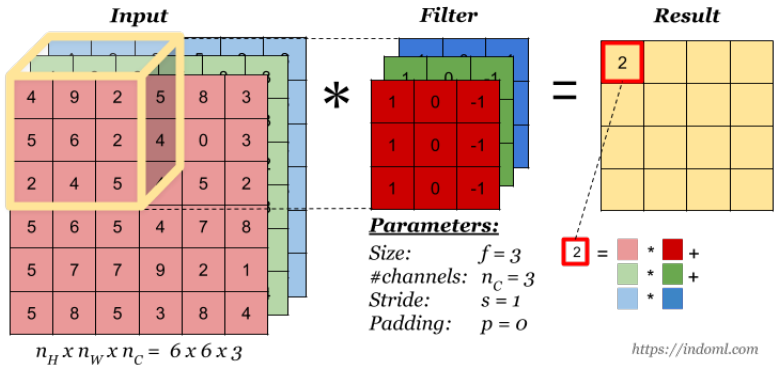


Figure 2.16: Convolution operation for 3 channels

A padding setting can also be used, which allows using a convolutional layer by preserving or changing the width and height of the feature map at the output of the layer. Moreover, this setting helps to save more information at the border of the image.

When the input includes several channels (see Figure 2.16), the filter should have the same number of channels as the input. In this case, convolution is performed on each matching channel, and then the result is added.

Eventually, a bias is added to the convolution layer, followed by an activation function, for example, ReLU or tanh. The deconvolution operation is the inverse of convolution.

Max-pooling layer

On the max-pooling layer, the filter is applied to the convolved feature maps. This operation is a downsampling operation and is used to reduce the size of the feature maps and the number of optimized parameters. Figure 2.17 shows the application of a 2x2 3D filter with a stride of 2 to the input image, the maximum value is returned from the each colored part of the image covered by the kernel. The opposite operation, known as unpooling, is also feasible.

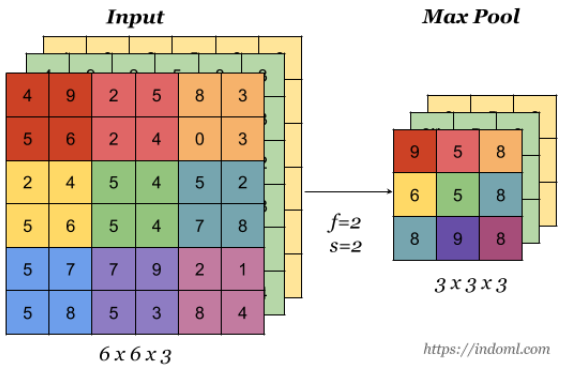


Figure 2.17: Max-pooling for input with multiple channels

Activation function

A common feature that allows different types of neural networks to learn complex tasks is the usage of non-linear activation functions. Indeed, any deep neural network would reduce to a simple regression model with just a linear activation function. There are several standard choices for the activation function: sigmoid, tangent hyperbolic, softmax (for multi-class problems), rectified linear activation function (ReLu), LeakyRelu (Fig.2.13). Of course, different activation functions work better for different tasks. Historically, the first choice of the activation function was the sigmoid function, used for the seminal perceptron network. However, the problem with it is that its slope is almost horizontal for large values of the arguments, and hence the problem learns effectively only for small values of arguments. This feature causes the so-called vanishing gradient problem, which is especially profound for deep networks with too many hidden layers (in more detail, vanishing gradient problem occurs when an argument of the activation function is a large number. For instance, for the sigmoid activation function the derivative is almost zero for such values of arguments, and this makes the derivatives of the loss function to be very close to zero; this results in the algorithm to work very slow, since it is hard for the changes to backpropagate through the layers.)

The common choice for the activation function in convolutional layers is the ReLU function, which has a positive slope for all positive values of arguments. For the last fully connected layer (also called the output layer), the natural choice is the sigmoid (for two-class problems) or softmax activation function (for multi-class problems). The task of convolutional layers is to learn local spaced features (low-level features at the shallow layers and more detailed at deeper layers). The task of fully connected layers is to combine local features learned at convolutional layers and combine them to make a prediction.

2.3.4 U-Net

The U-Net architecture is a convolutional neural network (CNN) that was created for a specific task of biomedical image segmentation by O. Ronneberger, P. Fischer, T. Brox in 2015 [45]. It is a modified and improved architecture of the fully convolutional network that was designed for semantic segmentation by E. Shelhamer, J. Long, T. Darrell in 2014 [46].

U-Net is a relatively deep convolutional neural network, and different layers extract different information about the image as described in the section of CNN. The deeper the convolutional network, the more likely a vanishing gradient problem will arise. Because the neural network is set up with random weights at the start, due to the vanishing gradient, the weights of the first layers cannot be updated, and they remain random. To avoid vanishing gradient, skip connections are introduced into this architecture (gray arrows in the Figure 2.18). These additional pathways are often helpful for model convergence, according to experiments. Skip connections jump certain layers in the neural network and feed the output of one layer as the input to the following layers through multiple layers rather than just the next one.

U-Net is used to perform the image-to-image translation (Section 3.1). Various generative adversarial neural networks involve U-Nets in a generator that carries out the actual part of image-to-image translation (Section 2.3.5).

As shown in the Fig.2.18, the U-Net consists of two parts: an encoding part and a decoding part, in other words, convolutional and deconvolutional layers, without fully connected layers. In the encoding part, applying the convolution operation and the max pooling operation, the image resolution is reduced at each subsequent level. During compression, spatial information is reduced and feature information enhances. The decoder part takes the feature information that the encoder has extracted and concatenates with the feature and spatial information obtained on the deconvolutional layers.

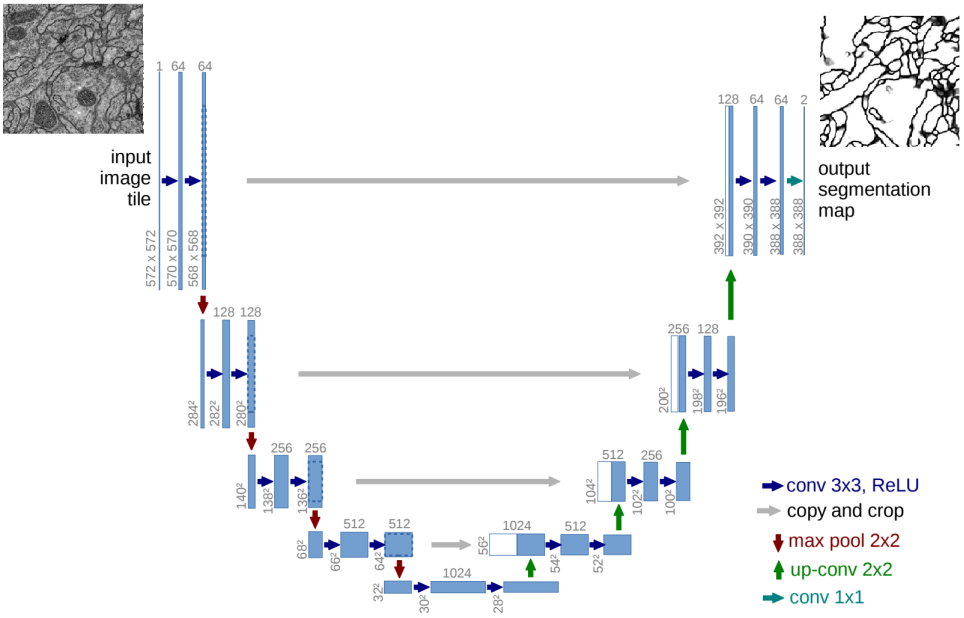


Figure 2.18: U-Net architecture. [45]

2.3.5 Generative Adversarial Network (GAN)

Since 2014, when generative adversarial networks (GANs) have been introduced [47], GANs have been extensively used in research [48], [49]. As a result, a significant variety of algorithms and applications that deal with image generation, image translation (converting one image to another) and other image tasks have been developed. Adversarial training is different kind of training than the supervised training, and it has proved to be quite effective in these tasks.

The architecture of generative adversarial networks involves two networks: the Generator (G) and the Discriminator (D) ($\mathcal{G}$ and $\mathcal{D}$ in Fig. 2.19, respectively).

During training, both Generator and Discriminator improve their performance by decreasing their own loss functions. When loss function for Generator goes down, the Discriminator loss function increases, and vice verse, hence the loss functions have nature. A typical choice for the loss function is mean square error (MSE).

The Generator (on the top of Figure 2.19) is in charge of generating a synthetic image by mapping a random input distribution to a particular data distribution. Then the generated image is passed as an input to the Discriminator, typically a binary classifier, which is in charge of judging whether that image is a real signal or a synthetic one. Subsequently, based on the Discriminator's decision, the Generator seeks to minimize loss between the real and the generated image, thereby generating images on each iteration more and more alike to the real input

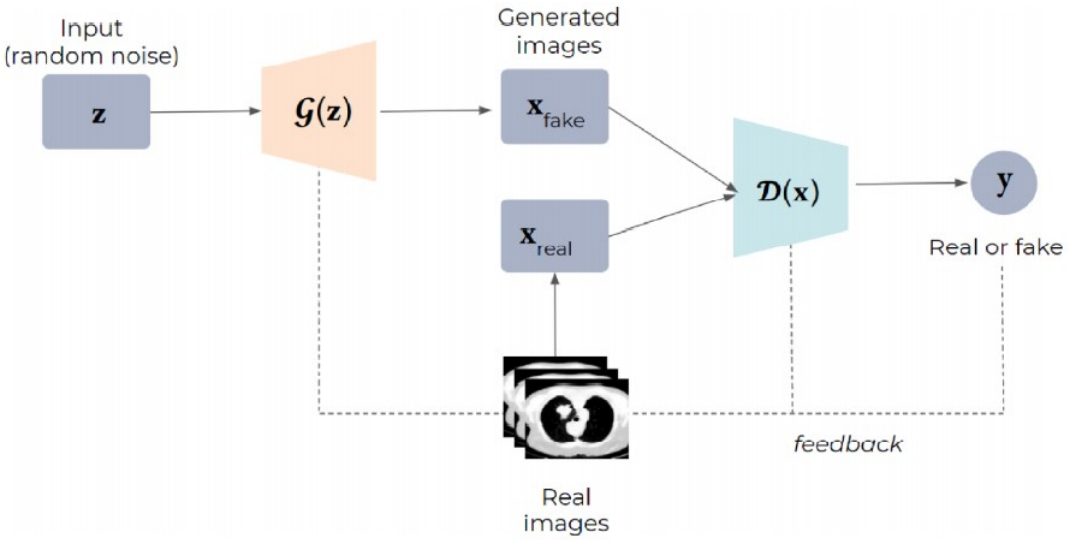


Figure 2.19: Generative Adversarial Network Credit:[13].

images (a typical measure of similarity of synthesized and real image is the mean squared error (MSE), which gives an averaged pixel-wise distance between two images). Eventually, the goal of Generator is to fool the Discriminator and convince it to classify the synthetic image as a real one. Moreover, at the same time, the Discriminator aims to learn better to identify generated data from real input data. Thus, the optimization of GAN is reduced to a minimax problem. Those two networks are trained jointly, and since they have opposite goals, one of the networks will sometimes be fixed, and the other one will be trained and then the other way around.

GANs have shown remarkable results in the tasks of image-to-image translation. For example, in Figure 2.20 a translation from zebra to horse and vice verse is demonstrated. We describe this in more detail in the next section.

Figure 2.20: Zebra-horse translation [56]

Chapter 3

Literature review

This section will talk about the most recent articles in image-to-image translation, including converting medical image modalities from CBCT to CT. We will discuss deep learning and classical methods without neural networks for converting CBCT to CT.

3.1 Image correction with convential techniques

To enhance the quality of CBCT images, various scatter correction approaches have been proposed over the past few years to fully integrate CBCT into ART workflow. These approaches are divided into hardware and software correction. First, a hardware correction approach is used to reduce scattering during the acquisition of projection data. So, by the time the detector absorbs the X-ray, its scattering is typically suppressed by specialized hardware (butterfly grid, anti-scattering grid, air gap and others); see [61] [60] for more detail. Nevertheless, the quantum nature of X-rays leads to the loss of important information and reduces the quality of an output image.

Attempts to deal with this problem were made by Onozato et al. [62] and Veiga et al. [63], who proposed a deformable image registration (DIR) to transform plan CT (pCT) into CBCT and which allows generating deformed plan CT (dpCT) with subsequent calculation of dose distribution. The DIR method is less suitable for soft tissues, such as the lung [64], pelvis [65, 66], where more complex anatomical changes might lead to incorrect calculation of dPCT dose distribution. Another method to resolve scattering correction is the Monte Carlo (MC) simulation [67]. An object is scanned multiple times to simulate the true path of all the photons. The pros of the method are its accuracy. However, the cons of this method are that it is very time consuming, which diminishes its clinical usefulness [68, 69].

Besides DIR and Monte Carlo, the analytical modelling method employs an

assumption that the scattering signal can be represented as a convolution of the main signal with the scattering kernel [70]. Another method is based on prior CT scans. It transforms the CT image and a CBCT image (computed with DIR from PCT) into an equally normalized uniform histogram. Taking CT image as a baseline, one then inverts the equalization operation on CBCT and thus corrects the differences between CT and CBCT, thus improving the quality of the CBCT image (improving density, energy stability, detector stability) [71]. This method, however, requires accurate paired PCT and CBCT datasets, which are hard to obtain in practice.

An extension of the latter method was done by Richter [72], who created a patient-based HU-D table, which allows to bypass PCT images and to compute radiation doses directly from CBCT. This method first divides the image into 2-6 regions with different tissue densities [73] and then calculate the doses based on the densities. This method, however, requires an accurate segmentation of an image into regions with different tissue categories.

3.2 Image conversion with deep learning

Over the past few years, there has been a growing number of deep convolutional networks applications to radiotherapy [54]. The potential advantages are their possible effectiveness and high robustness: learning the features on a big dataset once, they would be applied to other patients without modifications and give recommendations within few seconds. Unfortunately, those promising benefits are now mainly clouded by the low-quality images used in radiotherapy. Many studies are now concentrating on improving the quality of medical images.

All the up-to-date deep learning applications can be effectively divided into supervised learning and unsupervised learning. As for supervised learning, most of the research employs U-net and its modifications: Maier et al. [74] (eliminate artefact from CT and CBCT images), Kida et al. [51] (improve the uniformity of CBCT images, eliminate artefacts), Hansen et al. [52] (correct the CBCT dose), Li et al. [75] (U-net with residual blocks, improve quality of CBCT). These studies showed remarkable results, decreasing the root squared difference (RMSD) in fat and muscles tissues from 109 to 13 HU and from 57 to 11 HU, respectively [40], achieving 2% dose difference pass rate in radiotherapy and protontherapy treatments [76], and decreasing MAE between synthetic CT (sCT) and CT to about 20 HU. This research, however, required data used in the learning process to be paired, which is quite difficult to obtain, and which limits the application of the supervised learning technique.

Another technique is unsupervised learning, and it employs different types of generative adversarial networks (GANs): Wolterink et al. [77] (CycleGAN, convert

MRI of head and neck (H&N) to CT), Hiasa et al. [78] (GAN, convert MRI to CT in pelvic, introduced gradient consistency loss to obtain a good conversion effect), Liang et al. [58] (CycleGAN, synthesize CT from CBCT). Note that different organs require a different level of accuracy; for instance, the pelvic region is more challenging than the brain and H&N due to many organs at risk that are situated there.

Summing up, the advantages that make unsupervised learning based on different types of GANs a promising direction is that they allow potentially obtaining synthesized images of high quality (sCT) from a lower quality CBCT without any prior correspondence between CBCT and CT.

Chapter 4

Materials and Methodology

4.1 Materials

4.1.1 Data

The database we used consisted of 26 prostate patients. For training, we took 20 patients; for validation and testing, we took 1 and 5 patients, respectively (see Figure 4.1).

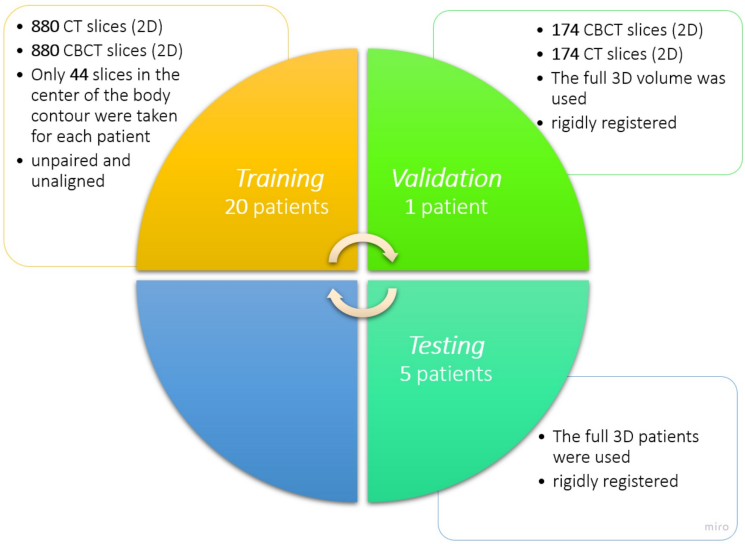


Figure 4.1: Splitting data into training, validation, and test sets.

Model training

For training, we focused on the CTV region, the region that will be irradiated (see Section 2.1.4) the slices containing the CTV region plus some more slices above and below up to 44. In total, we got 880 2D CT slices and 880 2D CBCT slices that were unpaired and unaligned.

For validation, we used the whole 3D CT and CBCT volumes of 1 patient (174 CBCT images and 174 CT images). They were rigidly registered in preprocessing stage (Fig.4.4) to calculate the mean absolute error (MAE) between the ground truth CT and synthesized CT volumes during the learning process.

Model evaluation

For testing, we took five patients whose CT and CBCT volumes were rigidly registered (Fig.4.4). We gave a CBCT as an input to the trained model, and as the output, we got a synthetic CT (sCT). Then, we computed the MAE between CT and sCT and the average of the MAEs over all test patients to evaluate the network's performance after training.

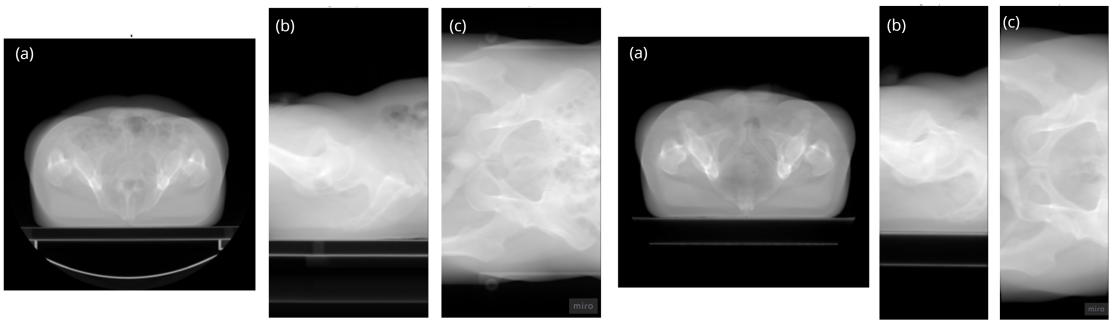


Figure 4.2: The example of original 3D CT volume (left) and CBCT volume (right) in the (a) axial, (b) sagittal and (c) coronal plane.

4.1.2 Raw data characteristics

- Original CT images had a matrix size of 500 by 500 on the axial plane and voxel intensity ranging from -1000 to above 3000. Therefore, the maximum HU value per implant is about 6300 (you can notice this specificity in Figure 2.5).
- Original CBCT images had a matrix size of 465 by 465 on the axial plane and the maximum HU value around 1500-1600 HU.

- Original CT and CBCT images were resampled to 1x1x1 mm using Reggui software and then saved in .mat format. Later, the conversion from .mat to NumPy arrays was performed.
- For the validation and test dataset, the rigid registration of the original CBCT and CT images was done using the commercial treatment planning system RayStation.
- Each CBCT and CT images had binary masks of the patient's body and internal pelvic organs such as the rectum, bladder and prostate.
- These patients had images that were taken with an interval of one or two weeks, so there could be some anatomical modifications in some patients, which implies that perfect ground truth CT was not available.

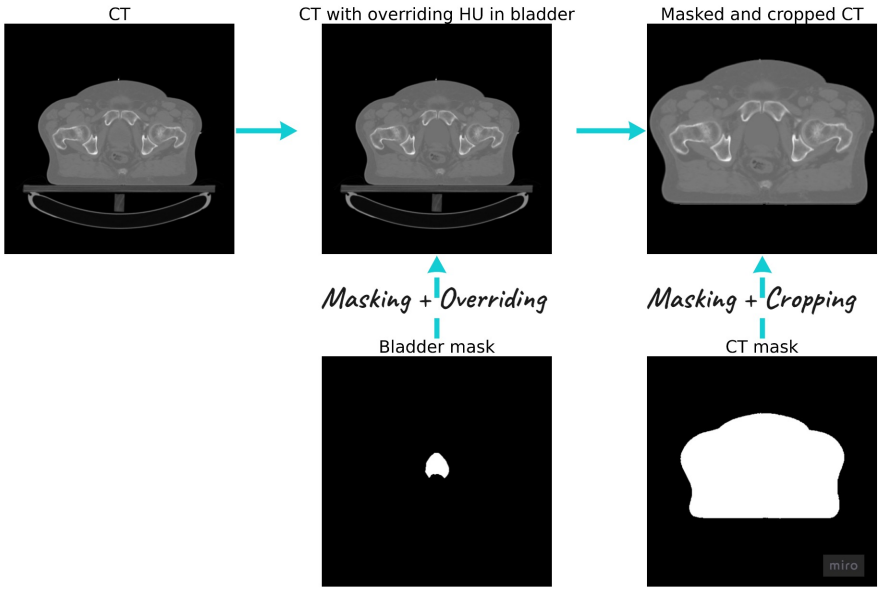


Figure 4.3: Preprocessing steps for training dataset showing on a single CT slice.

4.1.3 Data preprocessing

The data preprocessing for training data consisted of several steps (see Figure 4.3). First of all, the couch was removed on all training, validation and testing images. Since the couches from CBCT and CT are often different, we removed them to avoid bias. Therefore all voxels outside the body contour (binary mask) were fixed and set to -1000 HU (the third column in Figure 4.3).

Using a bladder mask (the second column in Figure 4.3), we override the values inside the bladder to 0 because some patients had a contrast agent in the bladder (see Figure 2.3 in Section 2.1.1). Contrast agent HU values are higher than the HU values of water that can lead to bias. The CT and CBCT images were cropped using a body mask, then cropped in the z-direction to get the only relevant part by organ location. For example, everything outside the organs masks was removed, just organs (bladder, prostate, rectum) were kept. Then initial voxel HU values were clipped to the range from -1000 and 1500 and normalized between 0 and 1.

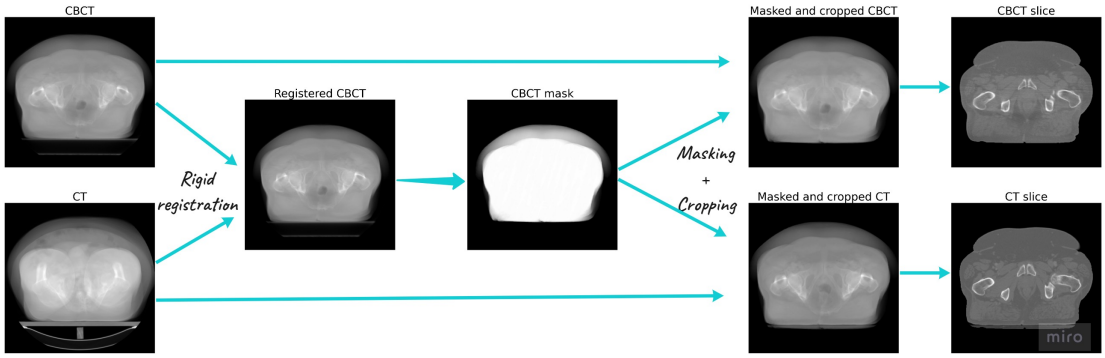


Figure 4.4: Preprocessing steps for validation and testing sets.

The original CBCT and CT images in the validation and test dataset were rigidly registered in the commercial treatment planning system RayStation, masking and cropping was performed (see Figure 4.4). In the last preprocessing step, all training, validation and test images were resampled to the desired size 256 by 256 on the axial direction.

4.1.4 Additional preprocessing

The raw data of some test patients had an external structure in the form of dots outside the body on CBCT scans but not on CT scans (Fig.4.5(a) left)

In order to eliminate bias in the test images, we decided to apply the multidimensional binary opening technique with a structuring element. The opening by the structuring element is a morphological image processing obtained by applying an erosion followed by a dilation. The noise produced by adding white dots to the image is called salt noise. It can be removed from the image using first erosion. Then, those areas that have survived the erosion with a given structuring element are restored to their original size due to dilation. This technique, allowed getting rid of the dots on the body mask while maintaining the shape of the primary body mask. In addition, we empirically found a suitable size of the structuring element, which was 15x15x1 for a 3D total volume (Fig.4.5(a)).

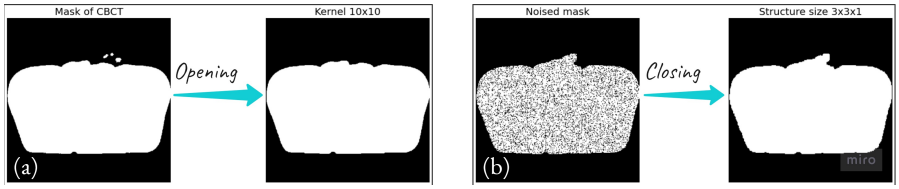


Figure 4.5: The morphological methods: (a) opening to set all pixels outside the binary body mask of CBCT to False and (b) closing to set all pixels inside the binary body mask of CBCT and CT to True

.

Another problem that has arisen when using binary mask sampling is another kind of noise, such as pepper noise ((Fig.4.5(b))). We used the closing technique using 3D structural element 3x3x1 (in morphology, the closing of a binary image by a structuring element is the erosion of the dilation of that image).

4.2 Method

To generate synthetic CT, we used the cycle-consistent adversarial neural network CycleGAN that performs the unpaired image-to-image translation. The code was inspired by the TensorFlow tutorial of CycleGAN for zebra-to-horse translation [80] and pix2pix framework [81]. From the initial architecture, we changed the number of channels to 1 as we used grayscale medical images, while in the tutorial, three-channel were used for RBG images.

4.2.1 Neural network architecture

The architecture of CycleGAN

The architecture of the entire CycleGAN network is depicted in the flow chart in Figure 4.6. CycleGAN is composed of two parts: *CycleF* and *CycleB*, which together constitute a forward and backward cycle. On top left in Figure 4.6 the forward cycle *CycleF* denotes the conversion from CBCT to CT, whereas on top right in Figure 4.6 the backward cycle *CycleB* signifies the reversal. CycleGAN also involves two discriminators (D_{CBCT} and D_{CT}) as well as two generators ($G_{CBCT-CT}$ and $G_{CT-CBCT}$). CycleGAN network was trained to convert CBCT (input) to synthetic CT (output).

Training of the generator in the part *CycleF* aims to create fake CT (synthetic CT, sCT) images from CBCT images that are as similar to the real CT as much as possible. The discriminator's training goal is to differentiate between the real CT images and the synthetic CT images generated by the generator. Ideally, the

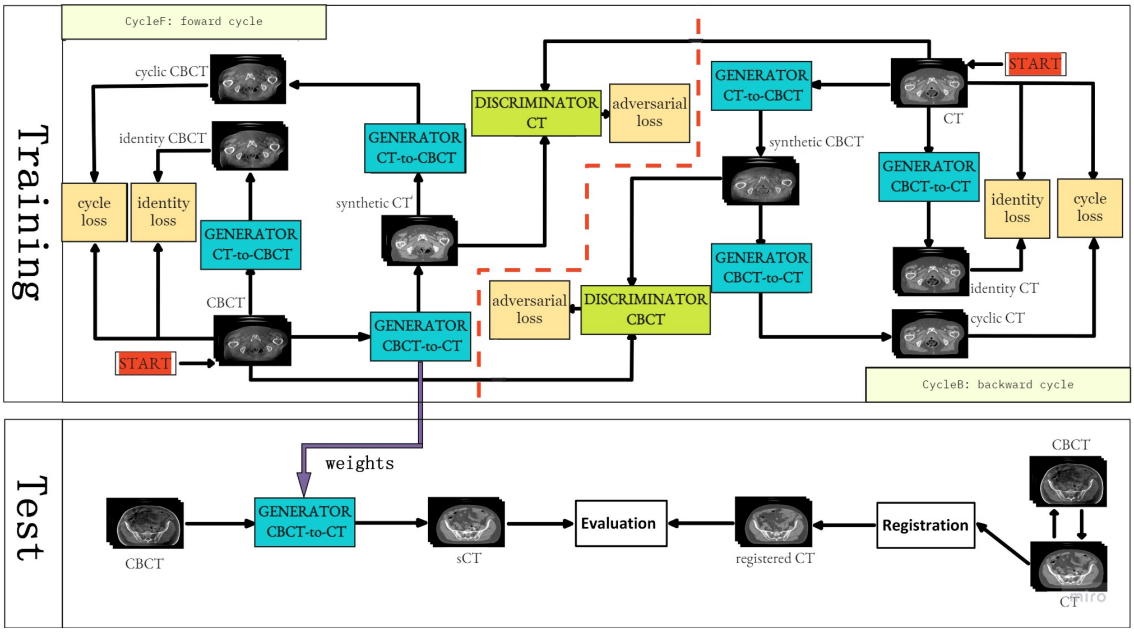


Figure 4.6: The schematic CycleGAN workflow for the image synthesis

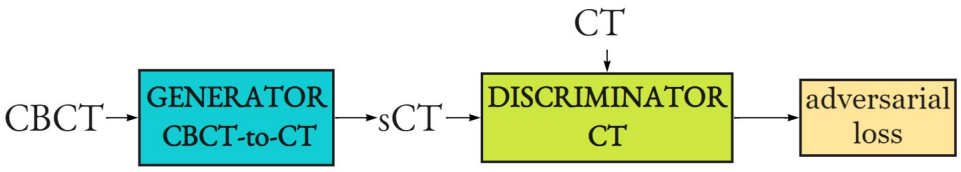
discriminator D_{CT} determines and outputs the labels 0 for the sCT images and 1 for the real CT images.

During the training, these two networks compete against each other, and they need to be regularly improved to make the sCT images as similar to the CT images as feasible. It means that the discriminator must acquire a more powerful classification ability to balance the competition.

Loss functions

For solving the optimization problem, three types of conditions were introduced in the loss function: adversarial loss L_{adv}, cycle-consistency loss L_{cycle}, and identity loss L_{iden} [56].

- **Adversarial loss**



The adversarial loss function L_{adv} aims to match generated images with target images [47]. Thus, using an adversarial loss function improves the

performance of the generator and discriminator networks. Practice shows that CycleGAN has challenges with stability during training. Therefore, to make the training process without fluctuations and more stable, we employed the least-squares loss function in the adversarial loss as it was proposed in LSGAN [57]. Thus, adversarial losses were used for generator G and its discriminator D and were written as follows:

Forward cycle.

$$L_{advD_{CT}} = \frac{1}{m}\sum_{i=1}^{m}\frac{(1 - D_{CT}(CT_i))^2 + D_{CT}(G_{CBCT-CT}(CBCT_i))^2}{2}, \quad (4.1)$$

$$L_{advG_{CBCT-CT}} = \frac{1}{m}\sum_{i=1}^{m}(1 - D_{CT}(G_{CBCT-CT}(CBCT_i)))^2, \quad (4.2)$$

where $G_{CBCT-CT}$ exerts to generate images $G_{CBCT-CT}(CBCT)$ (= sCT) so that they are identified by D_{CT} as real CT images with label 1 (minimization of $L_{advG_{CBCT-CT}}$), and D_{CT} strives to classify synthetic images sCT with label 0 and real CT images with label 1 (minimization of $L_{advD_{CT}}$).

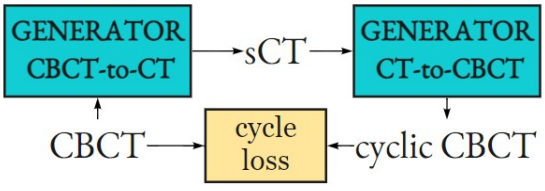
Backward cycle.

$$L_{advD_{CBCT}} = \frac{1}{m}\sum_{i=1}^{m}\frac{(1 - D_{CBCT}(CBCT_i))^2 + D_{CBCT}(G_{CT-CBCT}(CT_i))^2}{2}, \quad (4.3)$$

$$L_{advG_{CT-CBCT}} = \frac{1}{m}\sum_{i=1}^{m}(1 - D_{CBCT}(G_{CT-CBCT}(CT_i)))^2, \quad (4.4)$$

where the goals of $G_{CT-CBCT}$ and D_{CBCT} are the same as in forward cycle, but only for CBCT and synthetic CBCT.

- **Cycle-consistency loss**



After applying adversarial losses, the space of proper mapping functions is still large; for reducing this space, a cycle consistency loss was introduced, which minimized the difference between cyclic CBCT and real CBCT images,

cyclic CT and real CT images. Cycle-consistency losses for both cycles were as follows:

Forward cycle.

$$L_{cycle_{CBCT}} = \frac{1}{m}\sum_{i=1}^{m} |G_{CT-CBCT}(G_{CBCT-CT}(CBCT_i)) - CBCT_i|. \quad (4.5)$$

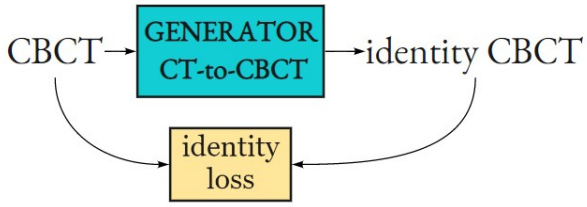
where $G_{CBCT-CT}$ generates sCT from CBCT image, while $G_{CT-CBCT}$ transforms sCT into cyclic CBCT (cCBCT).

Backward cycle.

$$L_{cycle_{CT}} = \frac{1}{m}\sum_{i=1}^{m} |G_{CBCT-CT}(G_{CT-CBCT}(CT_i)) - CT_i|. \quad (4.6)$$

where $G_{CT-CBCT}$ generates sCBCT from CT image, while $G_{CBCT-CT}$ transforms sCBCT into cyclic CT (cCT).

- **Identity loss**



The identity loss function is used to maintain the level of HU value between real CT and synthetic CT and between real CBCT and synthetic CBCT. For example, if $G_{CBCT-CT}$ takes a CT image as an input image instead of a CBCT image, the output will ideally be CT (identity CT). Moreover, the pixel-wise distance between the real CT and the identity CT is taken as the loss. And similarly for another generator. In other words, this loss function minimizes editing when going from CT to identity CT and from CBCT to identity CBCT without cycle. Therefore, the identity loss for both CBCT and CT images was :

Forward cycle.

$$L_{iden_{CT}} = \frac{1}{m}\sum_{i=1}^{m} |G_{CBCT-CT}(CT_i) - CT_i| \quad (4.7)$$

Backward cycle.

$$L_{iden_{CBCT}} = \frac{1}{m}\sum_{i=1}^{m} |G_{CT-CBCT}(CBCT_i) - CBCT_i|. \quad (4.8)$$

As a result, all losses were added and a total loss function for both generators was calculated using the formula:

$$L_{Gen} = L_{advG_{CBCT-CT}} + L_{advG_{CT-CBCT}} + \lambda_{cycle} \cdot (L_{cycle_{CT}} + L_{cycle_{CBCT}}) + \lambda_{iden} \cdot (L_{iden_{CT}} + L_{iden_{CBCT}}). \quad (4.9)$$

where the hyperparams λ_{cycle} is a parameter that regulates the weight of cycle-consistency loss and λ_{iden} regulates the weight of identity.

The final objective function for the discriminators was defined as:

$$L_{Disc} = L_{advD_{CT}} + L_{advD_{CBCT}}. \quad (4.10)$$

The architecture of the generator

For generators, we employed the modified U-Net architecture (Fig.4.7) with the encoder and decoder parts. The encoder consisted of convolutional layers, instance normalization and the activation function LeakyReLU.

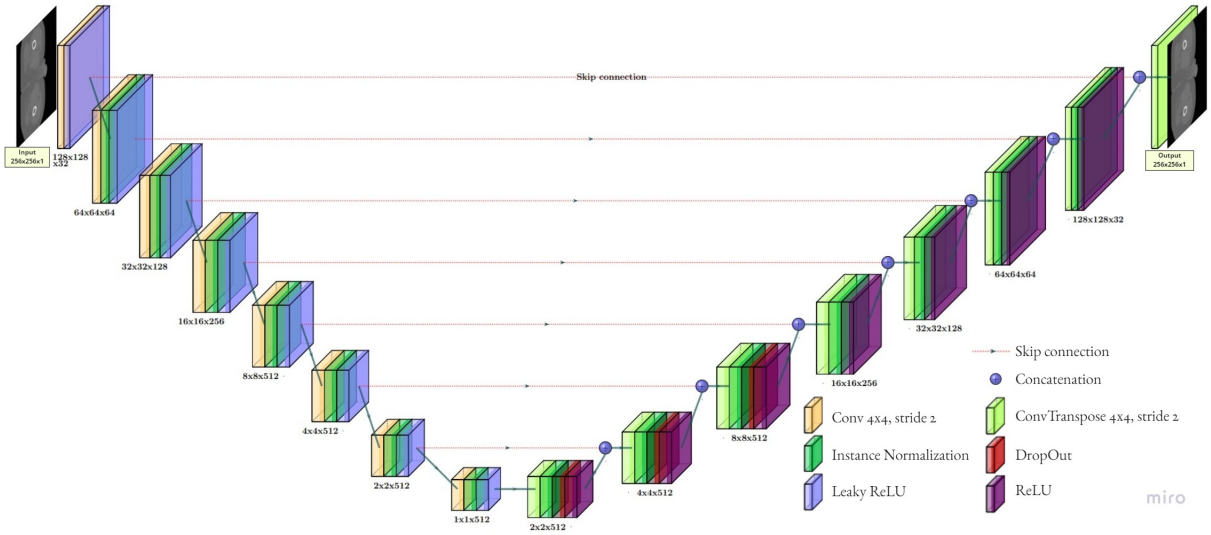


Figure 4.7: The architecture of the generators: modified U-Net. The size of input data and output data is 256 × 256 × 1 (where the first and second number is resolution of the image, the third number is the number of channels).

The decoder consisted of transpose convolutional layers, instance normalization, dropout for the first three blocks and ReLU as activation function. U-Net architecture had skip connections between encoder and decoder.

The architecture of the discriminator

The architecture for the discriminators was PatchGAN (Fig.4.8) [55]. The discriminator PatchGAN penalizes input structure at the level of local patches of the image.

PatchGAN consisted of four blocks with the same sequence of layers as the encoder in the generators: convolutional layers, instance normalization and the activation function LeakyReLU. After the third and fourth blocks, we used zero paddings that added zeros around the perimeter of the image tensor. The last layer was the convolutional layer with the stride of 1 and the number of filters of 1.

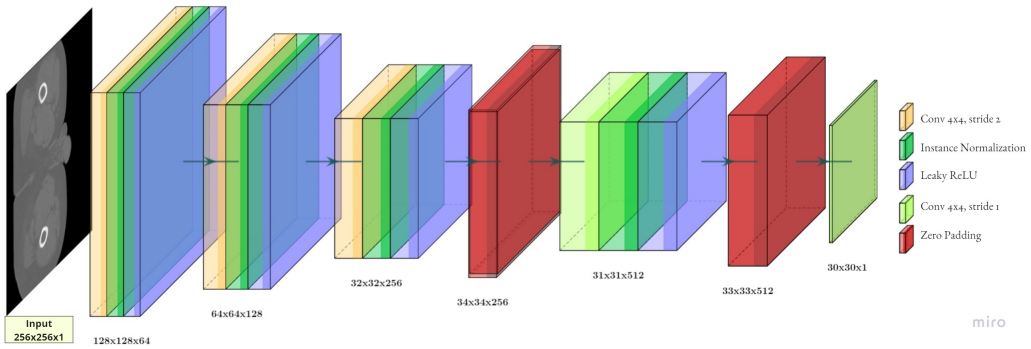


Figure 4.8: The architecture of the discriminators: PatchGAN. The size of input data is $256 \times 256 \times 1$.

4.2.2 Training process

In this section, we describe the training process and model training parameters. As described in the previous section, all given data were divided into 20 training (880 slices), 1 test (44 slices) and 5 test (220 slices).

Technical details

For this work used an NVIDIA TITAN XP GPU with 12GB of memory to train and test the model. One epoch took about 200 seconds for training and approximately 30 secs for testing.

The weights of the generators and discriminators were initialized with a random normal initializer (mean 0 and standard deviation 0.02). We used instance normalization at batch size 1. To stabilize the network training, we used the Adam optimization algorithm [82] with parameters $\beta_1 = 0.5, \beta_2 = 0.999$. At each stage of optimization, slices of different patients were randomly selected (see Table 4.1 in more detail of initial parameters).

Table 4.1: Initial selection of the hyperparameters of the model

Hyperparameters	**Initial selected values**
Number of U-Net filters	64, 128, 256, 512, 512, 512, 512, 512
Activation functions of U-Net	LeakyReLU, ReLU, Tanh
Number of PatchGAN filters	64, 128, 256, 512
Activation function of PatchGAN	ReLU
Normalization	Instance normalization
Batch size	1
Objective loss	Binary Cross Entropy
λ_{cycle}	100
$\lambda_{identity}$	100
Optimizer	Adam
learning rate	2e-4
learning rate mode	fixed for four networks
b_1, b_2	0.5, 0.99
Number of epochs	100

Objective loss

First we investigated binary cross entropy loss function in adversarial loss, but mean squared error (MSE) seemed to predict sCT with lower error (MAE) respect to the CT.

Learning rate

One of the important parts of training CycleGAN is to find an appropriate learning rate for four networks in CycleGAN. As shown in Tabel 4.1, initially, the learning rate parameters were set for each generator and discriminator to 2e-4. Correct hyperparameters are critical to training success, but they can be difficult to find. Although numerous experiences were done manually, it was still not clear which learning rate was required. Observations show that the generator is very sensitive to the learning rate of the discriminator. Many studies discuss the range from 2e-4 to 2e-6.

One approach to finding the optimal learning rate, grid search, requires full training with various learning rates. After complete training, the learning rate is chosen with the least final loss as a good and optimal learning rate. This repetitive network training requires a lot of resources and time.

We have implemented an approach to find a optimal learning rate based on a learning rate finder [83].

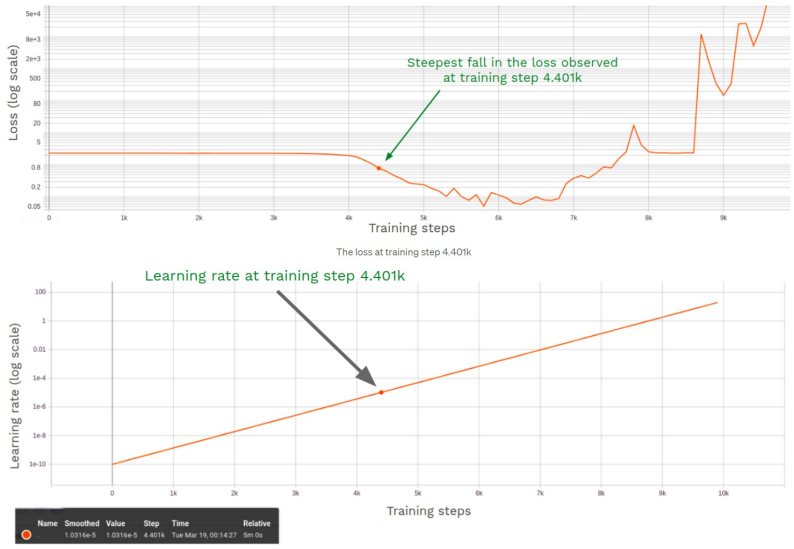


Figure 4.9: Learning rate finder. Starting from 1e-10 and increasing to 1e+2, we plot loss function, and we peeked the value of the learning rate at the point where the slope goes down [83].

With this approach, the network is trained only once. Thus, this method is cheaper in terms of resources and time than grid search. Nevertheless, in terms of reliability and robustness, LRFinder is inferior to grid search.

After finding the optimal learning rate for four networks, we set these values in the learning rate planner for the generator and the discriminator. We used exponential decay, polynomial, but the piece-wise function performed better. The piece function consisted of two parts. In the first part, the first few epochs, for example, 20-40 epochs, generators and discriminators have a fixed learning rate. In the second part, the learning rate decreases linearly to some specific value or zero.

4.2.3 Model validation

To quantify the generated images during training and for evaluation of the model in testing part, the mean absolute error metric between sCT and CT was used.

- Mean absolute error (MAE)

  MAE measures the difference between paired images, in our case synthetic CT and real CT images by calculating mean absolute error. The MAE is determined as follows:

$$MAE = \frac{1}{M}\sum_{i=1}^{n}|CT_i - sCT_i| \tag{4.11}$$

where n - the number of pixels in sCT and CT; M - the product number of pixels in sCT by number of pixels in CT (nxn)

4.2.4 Testing process

In this section, we describe the testing process and evaluation of the models using different metrics.

Figure 4.10 shows the testing flow chart. The testing dataset is rigidly registered for model evaluation using the commercial RayStation software to acquire paired CT and CBCT images.

The test CBCT images were fed to the generator $G_{CBCT-CT}$, into which the weights of the trained model were loaded. We used the obtained CT images as the ground truth for the quantitative assessment of the CycleGAN model.

Below in this section, we describe the metrics we used to compare sCT to CT quantitatively. For the synthesized image sCT, we measure the CT number accuracy (HU), structural similarity, and spatial uniformity.

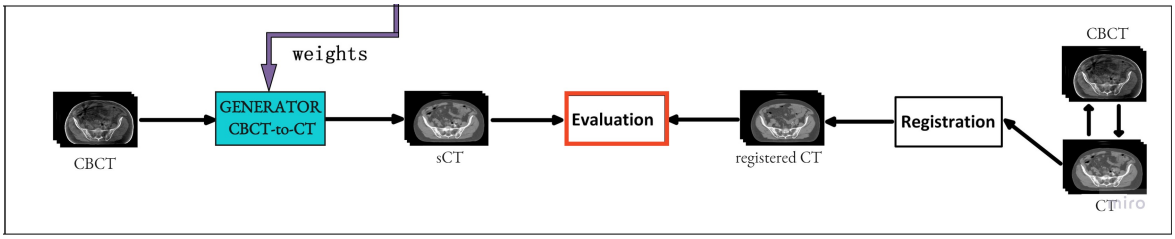


Figure 4.10: The schematic testing process.

Metrics

- Root mean square error (RMSE)

  RMSE measures the deviation between synthetic CT and real CT.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(CT_i - sCT_i)^2} \tag{4.12}$$

- Peak-signal-to-noise-ratio error (PSNR)

  PSNR is a metric for assessing of image denoising.

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX^2}{\frac{1}{n}\sum_{i=1}^{n}(CT_i - sCT_i)^2}\right) \tag{4.13}$$

- Structural similarity index measure (SSIM)

  SSIM is a structure information matrix used to compare the similarity of two images.

$$SSIM = \frac{(2\mu_{sCT}\mu_{CT} + c_1)(2\sigma_{sCT,CT} + c_2)}{(\mu_{sCT^2} + \mu_{CT^2} + c_1)(\sigma_{sCT^2} + \sigma_{CT^2} + c_2)} \tag{4.14}$$

  where n - the number of pixels in sCT (CT); MAX - the maximal pixel intensity in sCT and CT; μ - the mean value of the given image; σ - the standard variation of the given image, c_1 and c_2 - the variables which is used to keep a division with a weak denominator from becoming unbalanced.

We evaluated models by the mean average error (MAE) on the body contour. The proposed method achieved averaged MAE equal to 73 $\pm$ 20 HU, while with the initial parameters the model performance was at the level of 228 HU. Initial parameters of the model are listed in table 4.1, and the evolved parameters are listed in table 5.1. We will consider the experiments that led to the best MAE performance for our model in the next section.

Chapter 5

Results

Experiments

In this section we describe a sequence of different models on which we ran our experiments.

Table 5.1: Initial selection of the hyperparameters of the model

Experiments	**MAE**
Initial parameters	228.35 HU
U-Net Filters: 16, 32, 64, 128, 256, 512, 512, 512	215.40 HU
U-Net Filters: 32, 64, 128, 256, 512, 512, 512, 512 $lr_g = 2 \cdot 10^{-5}$ $lr_d = 2 \cdot 10^{-6}$	188.18 HU
$\lambda_{cycle} = 300$	171.17 HU
Loss: Mean Squared Error	156.33 HU
PatchGAN filters: 32, 64,128, 256	146.63 HU
Learning rate scheduler for generators, lr scheduler for discriminators	123.16 HU
The second dataset	79.17 HU
44 slices in the test data	73.84 HU

Initial model

The details of the initial model are presented in Table 4.1. As an objective function we took Binary Cross Entropy, Adam optimizer, the weights are initialized randomly with mean zero and standard deviation 0.02. The performance of the model is measured in MAE. The initial model had 228.35 HU.

Sequence of changes

After that we made a sequence of changes to the initial model, they are described in Table 5.1.

- First we changed the U-Net filters. As was described in Section 2.3.3, each layer of filters is aimed to determine some features and patterns; the number and size of filters depend on the complexity of an image and its details, and must be tune for each specific set of data. For simpler images one typically needs 3×3 or 4×4 kernels. Those filters distinguish local specificity of an image.

  Experiments showed that 32 filters in the first layer of the Generator is a good choice for our database.

- I also noted that increasing the value of λ_{cycle} leads to better results in HU values of MAE. This keeps anatomical structure of CBCT. For increasing the values of $lambda_{cycle}$ we started with the default value around 10, increased it up to 300, and that gave improvement in results (the shape of the CBCT is preserved, in other cases the sCT is smaller and the boundaries are slightly erased).

- Another important parameter which requires appropriate tuning is the learning rate; finding an optimal learning rate for Generator and Discriminator allows to obtain better results. Appropriate for us choices of the learning rate were in range from 2e-4 to 2e-6.

- We also experimented with different objective functions: instead of default Binary Cross Entropy we applied Least Square Loss, which allowed to significantly reduce the value of MAE: from 171 to 156 HU.

- We also tuned the number of filters in Discriminator. It turns out that decreasing the number of filters in the first layer of PatchGAN from 64 to 32 gave an improvement.

- Usage of learning rate scheduler allowed to decrease the value of MAE from 146 to 123 HU.

- Further, we decided to change the preprocessing of the our training set. Our initial preprocessing consisted of resizing and normalization; in the new version of preprocessing we applied binary masks to organs, and thus get rid of points outside the body. Thus we got images with more concentrated regions of interest: prostate and organs at risks (rectal and bladder). Such a change allowed to significantly reduce the value of MAE: from 123 to 79 HU. We also adjusted the test set to this cropping.

  Original CBCT volume has typically non-informative first and last slices (pieces of parts of body that we are not interested in). Thus we estimated only middle slices of the volume in our test evaluation, which allowed to decrease MAE to 73 HU.

- Change in Architecture This is the best value of MAE that we obtained during our learning processes. However, that model did not manage to visually suppress the artefacts in the images. Thus we decided to try another architecture of the Generator that was inspired by [58]. In our new Generator we used skip connections with skipped two layers (instead of 1 skipped layer originally used in U-Net). In the Encoder part, we alternated convolutional layers with 1 and 2 strides. In the Decoder part, we added upsampling layer before convolution (instead of dropout in the original U-Net); the convolution layers have stride 1.

  This new model gave MAE of 75 HU, which is a bit worse than in the previous model. However, visually, this model manages to suppress artefacts better than in all the other models that we used.

As a result of these changes, we decreased the MAE from 228.35 HU to 73 HU and obtained two models with the competitive performance. One has a lower HU values in MAE, but visually contains artefacts. The second model suppresses artefacts, but has a slightly bigger MAE. In the next Section we will assess qualitatively (visually) and quantitatively the results of these models performance.

Qualitative assessment of generated image quality

In Figure 5.1 and 5.2 we see visual quality for five patients. The upper line shows CBCT images, middle line is sCT images, and the third line is original CT images. We see that for the second model the synthesized CT has a better quality than for the first model. We see that both models synthesize images with preserved anatomical structure. The first model gives better value for MAE, however, the second one generates image with less artefacts.

All quantitative testing results obtained for all patients are shown in Tables 5.2 and 5.3. These tables refer to the assessment of each model and show the measured

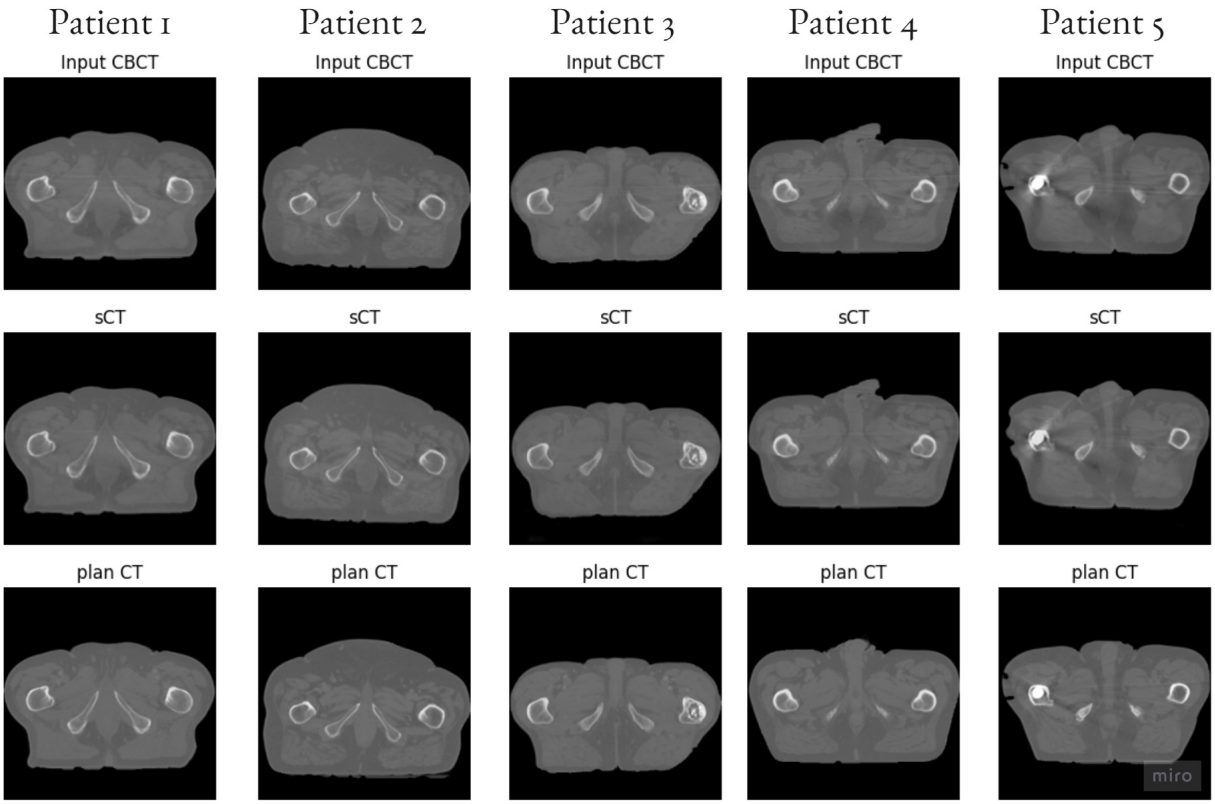


Figure 5.1: Model 1.

difference between the original CBCT and CT images, and the difference between synthetic CT and real CT.

The averaged MAE for the first model gives a bigger value than MAE for original CBCT and CT images (73±20HU in the model against 66± 20 HU in the original images) (see Table 5.2). However, in the test set, the fifth patient has an implant which even in the original images has MAE at the level 101 HU, which sufficiently increase the averaged MAE across all the patients.

The results for comparison of two models are shown in Table 5.4. In Table 5.5 we give values of SSIM metrics for CBCT-CT, sCT-CT, sCT-CBCT. In Figures 5.4, 5.6, and 5.7 we show HU profile of red dashed dotes to compare synthetic images for the first and second model.

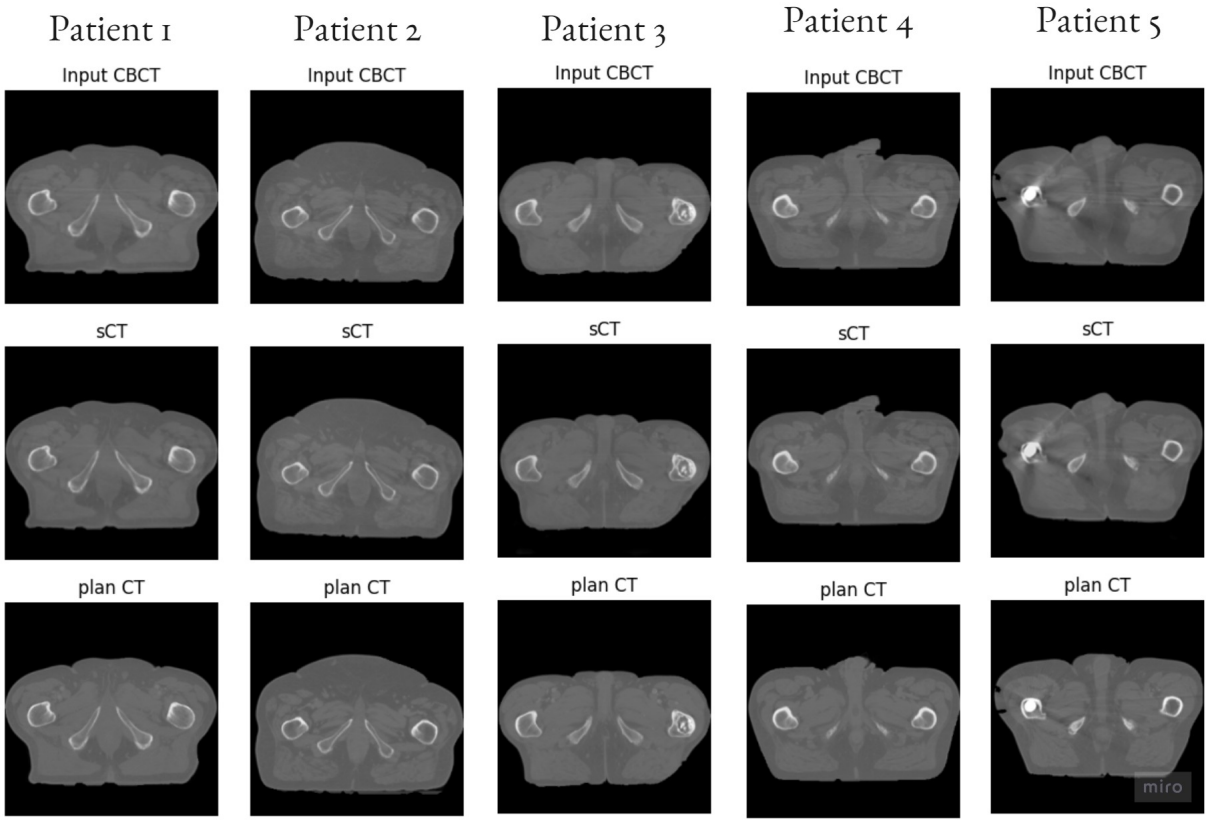


Figure 5.2: Model 2.

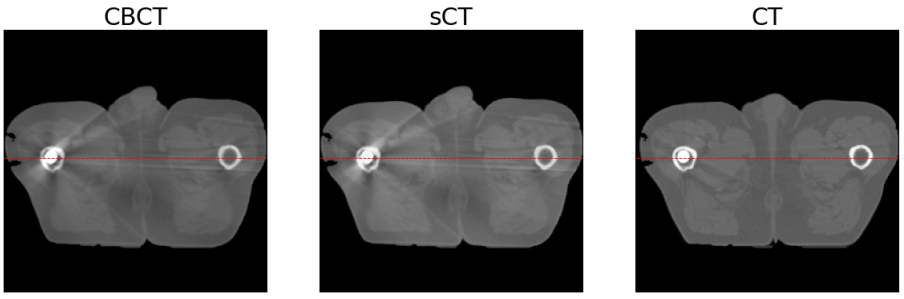


Figure 5.3: An example of CBCT image, sCT image synthesized from CBCT by CycleGAN, and planning CT (from the testing data). The red dashed line is a line profile that passes through bones, the implant and soft tissue.

Table 5.2: Measurements of difference CBCT - CT and sCT - CT in all 5 test patients for Model 1.

Patient	Difference	MAE (HU)	RMSE(HU)	SSIM	PNSE
Patient 1	CBCT-CT	68.87	157.04	0.90	24.04
	sCT-CT	79.79	157.06	0.90	24.04
Patient 2	CBCT-CT	67.01	133.99	0.90	25.42
	sCT-CT	67.53	133.41	0.89	25.46
Patient 3	CBCT-CT	46.09	96.42	0.95	28.28
	sCT-CT	54.89	97.73	0.95	28.16
Patient 4	CBCT-CT	47.18	89.19	0.96	28.95
	sCT-CT	56.26	93.66	0.96	28.53
Patient 5 (implant)	CBCT-CT	101.48	170.02	0.86	23.35
	sCT-CT	111.33	178.05	0.85	22.95
Mean ± std	CBCT-CT	66.13 ± 20.09	129.33 ± 32.06	0.91 ± 0.04	25.82 ± 2.24
	sCT-CT	73.96 ± 20.73	131.98 ± 32.85	0.91 ± 0.04	25.82 ± 2.21

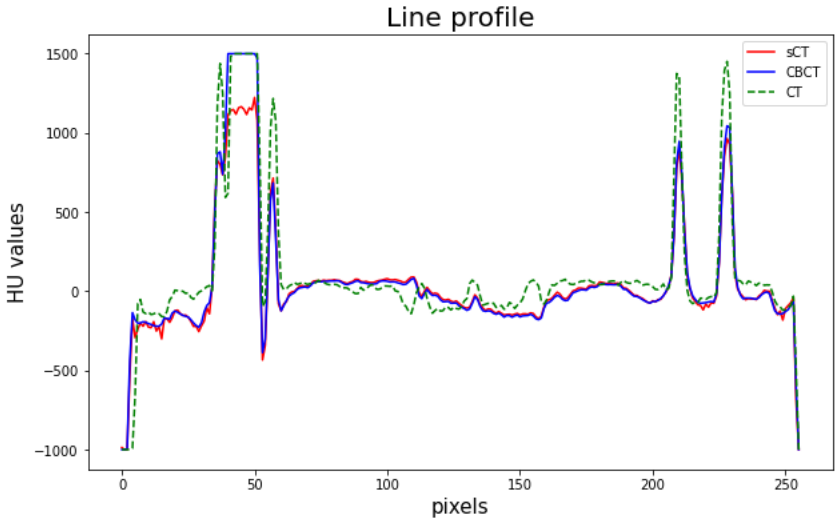


Figure 5.4: The plot shows HU profile of the red dashed lines in the Fig.5.5.

Table 5.3: Measurements of difference CBCT - CT and sCT - CT in all 5 test patients for Model 2

Patient	Difference	MAE (HU)	RMSE(HU)	SSIM	PNSE
Patient 1	CBCT-CT	68.87	157.04	0.90	24.04
	sCT-CT	75.31	155.83	0.90	24.11
Patient 2	CBCT-CT	67.01	133.99	0.90	25.42
	sCT-CT	78.98	163.45	0.87	23.69
Patient 3	CBCT-CT	46.09	96.42	0.95	28.28
	sCT-CT	60.31	115.36	0.94	26.72
Patient 4	CBCT-CT	47.18	89.19	0.96	28.95
	sCT-CT	57.20	122.86	0.95	26.17
Patient 5 (implant)	CBCT-CT	101.48	170.02	0.86	23.35
	sCT-CT	105.76	190.72	0.84	22.35
Mean ± std	CBCT-CT	66.13 ± 20.09	129.33 ± 32.06	0.91 ± 0.04	25.82 ± 2.24
	sCT-CT	75.51 ± 17.28	149.64 ± 27.60	0.90 ± 0.04	24.60 ± 1.61

Figure 5.5: The result of the second model. CBCT image, sCT image, and planning CT. The red dashed line is a line profile that passes through bones, the implant and soft tissue. The size of the window [-200, 500].

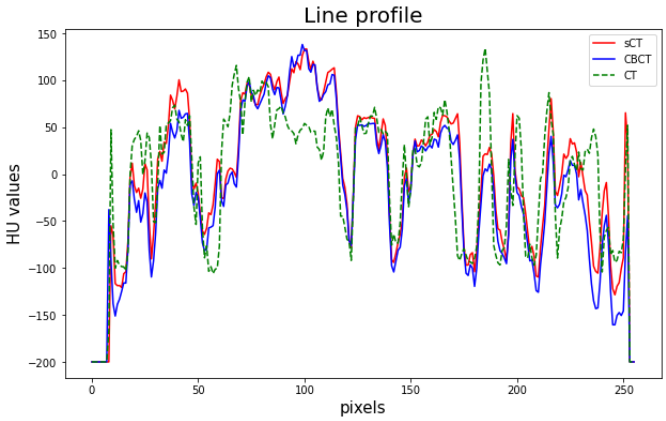


Figure 5.6: The plot shows HU profile of the red dashed lines in the Fig.5.5.

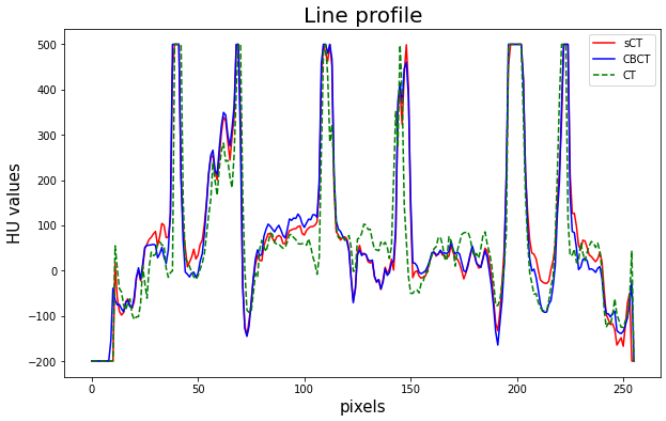


Figure 5.7: The plot shows HU profile of the red dashed lines in the Fig.5.5.

Table 5.4: Test data: CVT regions (including the patient with implant)

model	**MAE(HU)**	**Std**	**RMSE(HU)**	**SSIM**	**PNSE**
orig CBCT vs orig CT	66.13	± 20.09	129.33	0.91	26.01
model 1	73.96	± 20.73	131.98	0.91	25.82
model 2	75.51	± 17.28	149.65	0.90	24.61

Table 5.5: SSIM

Model	**Difference**	**SSIM ± std**	**Difference**	**SSIM ± std**	**Difference**	**SSIM ± std**
model 1	CBCT-CT	0.91 ± 0.04	sCT-CT	0.91 ± 0.04	sCT-CBCT	0.99 ± 0.001
model 2	CBCT-CT	0.91 ± 0.04	sCT-CT	0.90 ± 0.04	sCT-CBCT	0.90 ± 0.04

Chapter 6

Discussion and Conclusion

Here we will discuss the limitations of our work, possible ways to improve it, compare with existing literature, and talk about perspectives.

In the beginning of the project we aimed at being able to generate high-quality CT images from CBCT images. We did not fully achieve this initial results. Though our best model suppresses the artefacts for some patients, for the others it does not give desirable results. The state-of-the-art model gives MAE at the level of 20 HU, with the training process starting from 50 HU, while we started with MAE at the level of 66 HU and increased it to 73 HU.

There are several ideas how to improve the performance of our model, for instance to use residual blocks in the bottleneck of U-Net, or to schedule a more detailed tuning plan for the search of optimal hyperparameters. Besides, we noticed that preprocessing plays a crucial role in training of neural networks, and in some literature the authors use a rigid registration technique for training set, i.e. choose one image (in training data) as a reference, and align the rest of the images (both CBCT and CT images) along the reference, in order to get more accurate results. Though in principle CycleGAN does not require the alignment, it still might be crucial to improve its performance. It is also possible that our data have some defects, which prevented the model from better training. For instance, the first CBCT were taken up to two weeks after the planning CT, and therefore there could be anatomical modifications (such as air cavities, as an example) that can be a huge problem for the validation/testing of the model and might bias the generated results.

In our work we stayed at the image processing step. A possible interesting continuation of the work would be to perform the dose calculation on the sCT to see the impact of the image quality on the dose calculation.

Bibliography

[1] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, Ahmedin Jemal *Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries*, CA a Cancer Journal for Clinicians *DOI: 10.3322/caac.21492*, 2018

[2] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal,Freddie Bray *Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries*, CA a Cancer Journal for Clinicians *https://doi.org/10.3322/caac.21660*, 2021

[3] Brigid A. McDonald, Sastry Vedam, Jinzhong Yang, John Christodouleas, Clifton D. Fuller *Initial Feasibility and Clinical Implementation of Daily MR-Guided Adaptive Head and Neck Cancer Radiation Therapy on a 1.5T MR-Linac System: Prospective R-IDEAL 2a/2b Systematic Clinical Evaluation of Technical Innovation* DOI:*https://doi.org/10.1016/j.ijrobp.2020.12.015*, 2020

[4] *Understand Radiation Therapy Treatment for Brain Tumors*, BrainLab Org, URL: `https://www.brainlab.org/get-educated/brain-tumors/understand-radiation-therapy-treatment-for-brain-tumors/what-is-radiation-therapy/`, 2016

[5] Peter G. Hufton, Elizabeth Buckingham-Jeffery, Tobias Galla *Calculating normal tissue complication probabilities and probabilities of complication-free tumour control from stochastic models of population dynamics arXiv:1803.08595v1*, 2018

[6] Chris Wang *The progress of radiobiological models in modern radiotherapy with emphasis on the uncertainty issue*, PMID: 20178860, *DOI: 10.1016/j.mrrev.2010.02.001*, 2010

[7] Nevin McVicar, I. Antoniu Popescu, Emily Heath *Techniques for adaptive prostate radiotherapy*, DOI:*https://doi.org/10.1016/j.ejmp.2016.03.010*, 2016

[8] Qingrong Jackie Wu 1, Taoran Li, Qiuwen Wu, Fang-Fang Yin *Adaptive radiation therapy: technical components and clinical applications*, PMID: 21610472, DOI: *10.1097/PPO.0b013e31821da9d8*, 2011

[9] Jan-Jakob Sonke, Marianne Aznar, Coen Rasch *Adaptive Radiotherapy for Anatomical Changes*, DOI: *https://doi.org/10.1016/j.semradonc.2019.02.007 245*, 2019

[10] Yan D, Vicini F, Wong J, et al, *Adaptive radiation therapy.*, Phys Med Biol 42:123-132, https://www.ncbi.nlm.nih.gov/pubmed/9015813, 1997

[11] Michel Ghilezan, Di Yan, Alvaro Martinez *Adaptive Radiation Therapy for Prostate Cancer* Available online 26 February 2010. *https://doi.org/10.1016/j.semradonc.2009.11.007*, 2010.

[12] Wang T, Lei Y, Fu Y, Wynne JF, Curran WJ, Liu T, et al. *A review on medical imaging synthesis using deep learning and its clinical applications.* J Appl Clin Med Phys 2021;22:11–36. https://aapm.onlinelibrary.wiley.com/doi/10.1002/acm2.13121, 2021

[13] Ana Barragan-Montero, Umair Javaid, Gilmer Vald´es, Dan Nguyen, Paul Desbordes, Benoit Macq, Siri Willems, Liesbeth Vandewinckele, Mats Holmstrom¨, Fredrik Lofman¨,Steven Michiels, Kevin Souris, Edmond Sterpin , John A. Lee *Artificial intelligence and machine learning for medical imaging: A technology review*, DOI:https://doi.org/10.1016/j.ejmp.2021.04.016, 2021

[14] R.Schulze, U.Heil, D.Gross, D.D.Bruellmann, E.Dranischnikow, U.Schwanecke, E.Schoemer *Artefacts in CBCT: a review* PMID: 21697151 PMCID: PMC3520262 *DOI: 10.1259/dmfr/30642039*

[15] Lecturio, *Computed Tomography (CT)*, https://www.lecturio.com/concepts/computed-tomography-ct/, 2021

[16] Wikipedia, *Ct scan*, https://en.wikipedia.org/wiki/CT_scan, 2002-2021.

[17] Brenner DJ, Hall EJ, *Computed tomography – an increasing source of radiation exposure*, N. Engl. J. Med. 357 (22): 2277–84; DOI: *doi:10.1056/NEJMra072149*, PMID 18046031. Archived (PDF) from the original on 2016-03-04, 2007

[18] National Cancer Institute, NCI Dictionaries, *Spiral CT scan*, https://www.cancer.gov/publications/dictionaries/cancer-terms/def/spiral-ct-scan

[19] Jeffrey Klein, William E. Brant, Clyde Helms, Emily N. Vinson, *Brant and Helms' Fundamentals of Diagnostic Radiology* (Fifth ed.), Lippincott Williams and Wilkins, ISBN 978-1-4963-6738-9, 2018-07-19. Retrieved 24 January 2019.

[20] Shama Jaswal and Kevin M. Rice, *Prostatic Urethral Calculus*, Global Radiology CME, `https://www.globalradiologycme.com/single-post/prostatic-urethral-calculus`, Dec 7, 2020

[21] Shervin Kamalian, Rajiv Gupta *Handbook of Clinical Neurology*, 2016

[22] Orthopaedic Product News, *What is Cone Beam CT?*, `http://www.opnews.com/2017/05/what-is-cone-beam-ct/13533`, 2017

[23] Wikipedia, *Cone beam computed tomography*, `https://en.wikipedia.org/wiki/Cone_beam_computed_tomography`, 2011-2021.

[24] Jean Léger, Eliott Brion, Paul Desbordes, Christophe De Vleeschouwer, John A. Lee, and Benoit Macq. *Cross-domain data augmentation for deep-learning based male pelvic organ segmentation in cone beam ct.*, Applied Sciences, 10(3):1154, `http://hdl.handle.net/2078.1/226998`, 2020.

[25] Wikipedia, *Image registration*, `https://en.wikipedia.org/wiki/Image_registration`, 2002-2021.

[26] Guillaume Janssens, Laurent Jacques, Jonathan Orban de Xivry, Xavier Geets, and Benoit Macq, *Diffeomorphic Registration of Images with Variable Contrast Enhancement*, Int J Biomed Imaging. PMCID: PMC3005125, PMID: 21197460. Published online 2010 Dec 8. *doi:10.1155/2011/891585*, 2011.

[27] Andreas Wrangsjö, Johanna Pettersson, and Hans Knutsson *Non-Rigid Registration Using Morphons*, *doi:10.1007/11499145_51*, 2005.

[28] Meng Welliver, William T Yuh, Julia R Fielding, Katarzyna J. Macura, *Imaging across the Life Span: Innovations in Imaging and Therapy for Gynecologic Cancer*, Radiographics 34(4):1062-1081, DOI:10.1148/rg.344130099, 2014.

[29] Aurelien Geron *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'reilly, 2019.

[30] Mengke Qi, Yongbao Li, Aiqian Wu, Qiyuan Jia, Bin Li, Wenzhao Sun, Zhenhui Dai, Xingyu Lu, Linghong Zhou, Xiaowu Deng, Ting Song, *Multi-sequence MR image-based synthetic CT generation using a generative adversarial network for head and neck MRI-only radiotherapy*, Medical Physics, DOI:`https://doi.org/10.1002/mp.14075`, 2020.

[31] Paul F. Jaeger, Simon A. A. Kohl, Sebastian Bickelhaupt, Fabian Isensee, Tristan Anselm Kuder, Heinz-Peter Schlemmer, Klaus H. Maier-Hein *Retina U-Net: Embarrassingly Simple Exploitation of Segmentation Supervision for Medical Object Detection*, https://arxiv.org/abs/1811.08661v1, 2018.

[32] Hesamian MH, Jia W., He X., Kennedy P., *Deep learning techniques for medical image segmentation: achievements and challenges*, J Digit Imaging;32:582–96, 2019.

[33] Ibragimov B, Xing L. *Segmentation of organs-at-risks in head and neck CT images using convolutional neural networks*, Med Phys;44:547–57, 2017.

[34] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, et al. *Dermatologist-level classification of skin cancer with deep neural networks*, Nature, 542, pp. 115-118, doi: 10.1038/nature21056, 2017.

[35] R. Poplin, A.V. Varadarajan, K. Blumer, Y. Liu, M.V. McConnell, G.S. Corrado, et al. *Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning*, Nat Biomed Eng, 2, p. 158, 2018.

[36] Chen C, Dou Q, Chen H, Qin J, Heng PA., *Unsupervised bidirectional cross modality adaptation via deeply synergistic image and feature alignment for medical image segmentation*, IEEE Trans Med Imaging;39:2494–505, 2020.

[37] Perone CS, Ballester P, Barros RC, Cohen-Adad J. *Unsupervised domain adaptation for medical imaging segmentation with self-ensembling*, NeuroImage;194:1–11; DOI:https://doi.org/10.1016/j.neuroimage.2019.03.026., 2019.

[38] Yingzi Liu, Yang Lei, Tonghe Wang, Yabo Fu, Xiangyang Tang, Walter J. Curran, Tian Liu, Pretesh Patel and Xiaofeng Yang *CBCT-based Synthetic CT Generation using Deep-attention CycleGAN for Pancreatic Adaptive Radiotherapy. doi: 10.1002/MP.14121*, 2020.

[39] Lei Y, Harms J, Wang T, Liu Y, Shu H, Jani AB, et al. *MRI-only based synthetic CT generation using dense cycle consistent generative adversarial networks*, Med Phys;46:3565–81. DOI:https://doi.org/10.1002/mp.13617, 2019

[40] Aganj I, Harisinghani MG, Weissleder R, Fischl B. *Unsupervised medical image segmentation based on the local center of mass*, Sci Rep;8:13012, 2018.

[41] Peikari M, Salama S, Nofech-Mozes S, Martel AL. *A cluster-then-label semi-supervised learning approach for pathology image classification*, Sci Rep, DOI:https://doi.org/10.1038/s41598-018-24876-0, 2018.

[42] Ge C, Gu I-Y-H, Jakola AS, Yang J. *Deep semi-supervised learning for brain tumor classification*, BMC Med Imaging;20:87, 2020.

[43] Burton 2nd W, Myers C, Rullkoetter P. *Semi-supervised learning for automatic segmentation of the knee from MRI with convolutional neural networks*, Comput Methods Programs Biomed;189:105328, 2020.

[44] DeepMind, *MuZero: Mastering Go, chess, shogi and Atari without rules* `https://deepmind.com/blog/article/muzero-mastering-go-chess-shogi-and-atari-without-rules`, 2020.

[45] Olaf Ronneberger, Philipp Fischer, Thomas Brox *U-Net: Convolutional Networks for Biomedical Image Segmentation arXiv:1505.04597v1*, 2015.

[46] Jonathan Long, Evan Shelhamer, Trevor Darrell *Fully Convolutional Networks for Semantic Segmentation arXiv:1411.4038*, 2014.

[47] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio *Generative Adversarial Nets. arXiv:1406.2661v1*, 2014.

[48] Zhengwei Wang, Qi She, Tomas E. Ward *Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy. arXiv:1906.01529v4*, 2020.

[49] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, Jieping Ye *A Review on Generative Adversarial Networks Algorithms, Theory, and Applications. arXiv:2001.06937v1*, 2020

[50] Taasti et al, *Developments in deep learning based corrections of cone beam computed tomography to enable dose calculations for adaptive radiotherapy*, DOI:`https://doi.org/10.1016/j.phro.2020.07.012`, 2020.

[51] S. Kida, T. Nakamoto, M. Nakano, K. Nawa, A. Haga, J. Kotoku, et al. *Cone beam computed tomography image quality improvement using a deep convolutional neural network*, Cureus, 10, Article e2548, doi:10.7759/cureus.2548, 2018.

[52] D.C. Hansen, G. Landry, F. Kamp, M. Li, C. Belka, K. Parodi, et al., *ScatterNet: A convolutional neural network for cone-beam CT intensity correction*, Med Phys, 45, pp. 4916-4926, doi:10.1002/mp.13175, 2018.

[53] G. Landry, D. Hansen, F. Kamp, M. Li, B. Hoyle, J. Weller, et al., *Comparing U-Net training with three different datasets to correct CBCT images for prostate radiotherapy dose calculations*, Phys Med Biol, 64, Article 035011, doi:10.1088/1361-6560/aaf496, 2019.

[54] S. Xie, C. Yang, Z. Zhang, H. Li *Scatter artifacts removal using learning-based method for CBCT in IGRT system*, IEEE Access, 6, pp. 78031-78037, doi:10.1109/ACCESS.2018.2884704, 2018.

[55] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks*. arXiv preprint *arXiv:1611.07004*, 2017.

[56] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. arXiv preprint *arXiv:1703.10593*, 2017.

[57] Mao X, Li Q, Xie H, Lau RYK, Wang Z, Smolley SP. *Least Squares Generative Adversarial Networks*, Proceedings of the IEEE International Conference on Computer Vision 2794–2802. doi: 10.1109/ICCV.2017.304, 2017.

[58] Xiao Liang, Liyuan Chen, Dan Nguyen, Zhiguo Zhou, Xuejun Gu, Ming Yang, Jing Wang, Steve Jiang *Generating Synthesized Computed Tomography (CT) from Cone-Beam Computed Tomography (CBCT) using CycleGAN for Adaptive Radiation Therapy*. arXiv preprint *arXiv:1810.13350*, 2018.

[59] S.Kida, S.Kaji, K.Nawa, T.Imae, T.Nakamoto, S.Ozaki, T.Ohta,Y.Nozawa and K.Nakagawa. *Cone-beam CT to Planning CT synthesis using generative adversarial networks*. arXiv preprint *arXiv:1901.05773v1*, 2019.

[60] Liang X, Jiang Y, Xie Y, Niu T. *Quantitative Cone-Beam CT Imaging in Radiotherapy: Parallel Computation and Comprehensive Evaluation on the TrueBeam System*. IEEE Access, 7(6):66226–33. doi: 10.1109/ACCESS.2019.2902168, 2019.

[61] Liang X, Li N, Zhang Z, Yu S, Qin W, Li Y, et al. Shading *Correction for Volumetric CT Using Deep Convolutional Neural Network and Adaptive Filter*, Quantitative Imaging Med Surg (2019) 9(7):1242–54. doi: 10.21037/qims.2019.05.19

[62] Onozato Y, Kadoya N, Fujita Y, Arai K, Dobashi S, Takeda K, et al. Evaluation of on-Board Kv Cone Beam Computed Tomography–Based Dose Calculation With Deformable Image Registration Using Hounsfield Unit Modifications. Int J Radiat Oncol Biol Phys (2014) 89(2):416–23. doi: 10.1016/j.ijrobp.2014.02.007

[63] Veiga C, McClelland J, Moinuddin S, Lourenço A, Ricketts K, Annkah J, et al. Toward Adaptive Radiotherapy for Head and Neck Patients: Feasibility Study on Using CT-to-CBCT Deformable Registration for "Dose of the Day" Calculations. Med Phys (2014) 41(3):031703. doi: 10.1118/1.4864240

[64] Veiga C, Janssens G, Teng C-L, Baudier T, Hotoiu L, McClelland JR, et al. First Clinical Investigation of Cone Beam Computed Tomography and Deformable Registration for Adaptive Proton Therapy for Lung Cancer. Int J Radiat Oncol Biol Phys (2016) 95(1):549–59. doi: 10.1016/j.ijrobp.2016.01.055

[65] Kurz C, Kamp F, Park YK, Zöllner C, Rit S, Hansen D, et al. Investigating Deformable Image Registration and Scatter Correction for CBCT-Based Dose Calculation in Adaptive IMPT. Med Phys (2016) 43(10):5635–46. doi: 10.1118/1.4962933

[66] Giacometti V, King RB, Agnew CE, Irvine DM, Jain S, Hounsell AR, et al. An Evaluation of Techniques for Dose Calculation on Cone Beam Computed Tomography. Br J Radiol Suppl (2019) 92(1096):20180383. doi: 10.1259/bjr.20180383

[67] Naimuddin S, Hasegawa B, Mistretta CA. Scatter-Glare Correction Using a Convolution Algorithm With Variable Weighting. Med Phys (1987) 14(3):330–4. doi: 10.1118/1.596088

[68] Xu Y, Bai T, Yan H, Ouyang L, Pompos A, Wang J, et al. A Practical Cone-Beam CT Scatter Correction Method With Optimized Monte Carlo Simulations for Image-Guided Radiation Therapy. Phys Med Biol (2015) 60(9):3567. doi: 10.1088/0031-9155/60/9/3567

[69] Park YK, Sharp GC, Phillips J, Winey BA. Proton Dose Calculation on Scatter-Corrected CBCT Image: Feasibility Study for Adaptive Proton Therapy. Med Phys (2015) 42(8):4449–59. doi: 10.1118/1.4923179

[70] Boone JM, Seibert JA. An Analytical Model of the Scattered Radiation Distribution in Diagnostic Radiology. Med Phys (1988) 15(5):721–5. doi: 10.1118/1.596186

[71] Abe T, Tateoka K, Saito Y, Nakazawa T, Yano M, Nakata K, et al. Method for Converting Cone-Beam CT Values Into Hounsfield Units for Radiation Treatment Planning. Clin Eng R Oncol (2017) 6(4):361–75. doi: 10.4236/ijmpcero.2017.64032

[72] Richter A, Hu Q, Steglich D, Baier K, Wilbert J, Guckenberger M, et al. Investigation of the Usability of Conebeam CT Data Sets for Dose Calculation. (2008) 3(1):42. doi: 10.1186/1748-717X-3-42

[73] Barateau A, Perichon N, Castelli J, Schick U, Henry O, Chajon E, et al. A Density Assignment Method for Dose Monitoring in Head-and-Neck Radiotherapy. Strahlenther Onkol (2019) 195(2):175–85. doi: 10.1007/s00066-018-1379-y

[74] Maier J, Sawall S, Knaup M, Kachelrieß M. Deep Scatter Estimation (DSE): Accurate Real-Time Scatter Estimation for X-Ray CT Using a Deep Convolutional Neural Network. J Nondestr Eval (2018) 37(3):57. doi: 10.1007/s10921-018-0507-z

[75] Li Y, Zhu J, Liu Z, Teng J, Chen L. A Preliminary Study of Using a Deep Convolution Neural Network to Generate Synthesized CT Images Based on CBCT for Adaptive Radiotherapy of Nasopharyngeal Carcinoma. Phys Med Biol (2019) 64(14):145010. doi: 10.1088/1361-6560/ab2770

[76] Hansen DC, Landry G, Kamp F, Li M, Belka C, Parodi K, et al. Scatternet: A Convolutional Neural Network for Cone-Beam CT Intensity Correction. Med Phys (2018) 45(11):4916–26. doi: 10.1002/mp.13175

[77] Wolterink JM, Dinkla AM, Savenije MHF, Seevinck PR, van den Berg CAT, Isgum I. Deep MR to CT Synthesis Using Unpaired Data. In: International Workshop on Simulation and Synthesis in Medical Imaging. Springer (2017). p. 14–23. doi:10.1007/978-3-319-68127-6_2

[78] Hiasa Y, Otake Y, Takao M, Matsuoka T, Takashima K, Prince JL, et al. Cross-Modality Image Synthesis From Unpaired Data Using CycleGAN: Effects of Gradient Consistency Loss and Training Data Size. In: International Workshop on Simulation and Synthesis in Medical Imaging. Springer (2018). p. 31–41. doi:10.1007/978-3-030-00536-8_4

[79] Matteo Maspero, Antonetta C. Houweling, Mark H.F. Savenije, Tristan C.F. van Heijst,Joost J.C. Verhoeff, Alexis N.T.J. Kotte, Cornelis A.T. van den Berg *A single neural network for cone-beam computed tomography-based radiotherapy of head-and-neck, lung and breast cancer. https://doi.org/10.1016/j.phro.2020.04.002*, 2020.

[80] Tensorflow Tutorials: CycleGAN, https://www.tensorflow.org/tutorials/generative/CycleGAN

[81] Tensorflow Tutorials: pix2pix: Image-to-image translation with a conditional GAN, https://www.tensorflow.org/tutorials/generative/pix2pix

[82] Kingma DP, Ba J. *Adam: A Method for Stochastic Optimization*, arXiv preprin, 1412:6980, 2014.

[83] Medium. Octavian: *How to use the Learning Rate Finder in TensorFlow*, https://medium.com/octavian-ai/

how-to-use-the-learning-rate-finder-in-tensorflow-126210de9489, 2019.

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