

Deep learning for organ segmentation in radiotherapy: federated learning, contour propagation, and domain adaptation



Elliott Brion

Institute of Information and Communication
Technologies, Electronics and Applied Mathematics
(ICTEAM)

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor in Applied Sciences

February 22, 2020

PhD committee

Thesis supervisors

Prof. Benoit Macq

Institute of Information and Communication Technologies, Electronics
and Applied Mathematics, Université catholique de Louvain
École polytechnique de Louvain, Université catholique de Louvain

Prof. John A. Lee

Molecular Imaging, Radiotherapy and Oncology, Université catholique
de Louvain
École polytechnique de Louvain, Université catholique de Louvain

President of the jury

Prof. Jean-Pierre Raskin

Institute of Information and Communication Technologies, Electronics
and Applied Mathematics, Université catholique de Louvain
École polytechnique de Louvain, Université catholique de Louvain

Members

Prof. Christophe De Vleeschouwer

Institute of Information and Communication Technologies, Electronics
and Applied Mathematics, Université catholique de Louvain
École polytechnique de Louvain, Université catholique de Louvain

Prof. Xavier Geets

Institut Roi Albert II, Radiothérapie oncologique

Dr. Rudi Labarbe

Ion Beam Applications SA (IBA), Louvain-la-Neuve

External members

Prof. Romain Hérault

LITIS, INSA Rouen, Normandie Université

Prof. Bernard Gosselin

Université de Mons

Abstract

External radiotherapy treats cancer by pointing a source of radiation (either photons or protons) at a patient who is lying on a couch. While it is used in more than half of all cancer patients, this treatment suffers from two major shortcomings. First, the target sometimes receives less radiation dose than prescribed, and healthy organs receive more of it. Although some dose to healthy organs is inevitable (since the beam must enter the body), part of it is due to poor management of anatomical variations during treatment. As a consequence, the tumor can fail to be controlled (possibly leading to decreased quality of life or even death) and secondary cancers can be induced in the healthy organs. Second, the slowness of treatment planning escalates healthcare costs and reduces doctors' face-to-face time with their patients.

Coupled with steady improvement in the quality of the medical images used for treatment planning and monitoring, deep learning promises to offer fast and personalized treatment for all cancer patients sent to radiotherapy. Over the past few years, computation capabilities, as well as digitization and labeling of images, have been increasing rapidly. Deep learning, a brain-inspired statistical model, now has the potential to identify targets and healthy organs on medical images with unprecedented speed and accuracy. This thesis focuses on three aspects: slice interpolation, CBCT transfer, and multi-centric data gathering.

The treatment planning image (called computed tomography, or CT) is volumetric, i.e., it consists of a stack of slices (2D images) of the patient's body. The current radiotherapy workflow requires contouring the target and healthy organs on all slices manually, a time-consuming process. While commercial suites propose fully automated contouring with deep learning, their use for contour propagation remains unexplored.

In this thesis, we propose a semi-automated approach to propagate the contours from one slice to another. The medical doctor, therefore, needs to contour only a few slices of the CT, and those contours are automatically propagated to the other slices. This accelerates treatment planning (while maintaining acceptable accuracy) by allowing neural networks to propagate knowledge efficiently.

In radiotherapy, the dose is not delivered at once but in several small doses called *fractions*. The poorly measured anatomical variation between fractions (e.g., due to bladder and rectal filling and voiding) hampers dose conformity. This can be mitigated with the Cone Beam CT (CBCT), an image acquired before each fraction which can be considered a low-contrast CT. Today, targets and organs at risk can be identified on this image with registration, a model making assumptions about the nature of the anatomical variations between CT and CBCT. However, this method fails when these assumptions are not met (e.g., in the case of large deformations). In contrast, deep learning makes few assumptions. Instead, it is a flexible model that is calibrated on large databases. More specifically, it requires annotated CBCTs for training, and those labels are time-consuming to produce. Fortunately, large databases of contoured CTs exist, since contouring CTs has been part of the workflow for decades. To leverage such databases we propose *cross-domain data augmentation*, a method for training neural networks to identify targets and healthy organs on CBCT using many annotated CTs and only a few annotated CBCTs. Since contouring a few CBCTs may already be challenging for some hospitals, we investigate two other methods – *domain adversarial networks* and *intensity-based data augmentation* – that do not require any annotations for the CBCTs. All these methods rely on the principle of sharing information between the two image modalities (CT and CBCT).

Finally, training and validating deep neural networks often requires large, multi-centric databases. These are difficult to collect due to technical and legal challenges, as well as inadequate incentives for hospitals to collaborate. To address these issues, we apply *TCLearn*, a federated Byzantine agreement framework, to our use-case. This framework is shown to share knowledge between hospitals efficiently.

Acknowledgments

Most theses are too large enterprises to be completed by any single individual. I would like to express my gratitude to the people who supported me along the journey.

First, my family. Maman, papa, Elsa, Mamy, Boris, thank you for believing in me.

Second, my supervisors. Benoit, thank you for having trusted me four years ago and ever since. I appreciated your invariable enthusiasm and the opportunities you provided, including joining you for one month during your sabbatical at McGill University. John, thank you for your availability and precious advice.

Then come the friends. Corentin, Christophe, Adrien, and Gaël, thank you for having supported me during the lows and for having celebrated the highs.

My lab mates, a.k.a. “Team jprod”. Jean, there is no one else I would rather have shared the seat with during the emotional roller-coaster ride of this research project. Paul, Umair, Antoine, Sylvain, Gaetan, Damien, Ana, and Simon, thank you for having made this lab a fun, supporting, and stimulating place to be.

Thanks to my housemates Nicolas, Corentin, Laury, and Annelise, for our “repas coloc” and other activities providing precious relaxing time.

Without data, no deep learning. I would like to thank our hospital partners who trusted us with theirs: Dr. Jean-François Daisne and Dr. Vincent Remouchamps, from CHU-UCL-Namur, Nicolas Meert, from CHU-Charleroi, as well as the teams of doctors and physicians from both centers who welcomed us for several weeks.

I am not best known for my ability at handling administrative stuff. Patricia, thank you for your patience. Similarly, thank you to Brigitte and François, UCLouvain's system administrators, for your support.

This work was made possible thanks to Sara, who annotated a large amount of data used in the studies, and Gabrielle, who carefully revised this manuscript. Thanks to the two of you.

I also had the chance to be followed by a thesis committee of bright and helping scientists. Rudi, thank you for guiding and inspiring me from my internship at IBA when I was still a master's student to editing the present document. Christophe, thank you for your numerous ideas and guidance. Several chapters of this document have been greatly improved thanks to your valuable feedback.

Finally, I had the chance to start this Ph.D. with an inspiring internship at IBM Almaden. Mehdi and Hongzhi, thank you for your guidance on-site and your invitation to social activities offsite. This stay helped to put me on the right track.

List of publications

Related papers in peer-reviewed journals and conference proceedings.

Contour propagation in CT scans with convolutional neural networks

[102] Léger, J., Brion, E., Javaid, U., Lee, J., De Vleeschouwer, C., Macq, B. (2018, September). In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 380-391). Springer, Cham.

Using planning CTs to enhance CNN-based bladder segmentation on cone beam CT

[18] Brion, E., Léger, J., Javaid, U., Lee, J., De Vleeschouwer, C., Macq, B. (2019, March). Using planning CTs to enhance CNN-based bladder segmentation on cone beam CT. In Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling (Vol. 10951, p. 109511M). International Society for Optics and Photonics.

Secure architectures implementing trusted coalitions for blockchained distributed learning (TCLearn)

[112] Lugan, S., Desbordes, P., Brion, E., Tormo, L. X. R., Legay, A., Macq, B. (2019). Secure architectures implementing trusted coalitions for blockchained distributed learning (TCLearn). IEEE Access, 7, 181789-181799.

Cross-domain data augmentation for deep learning based male pelvic organ segmentation in cone beam CT

[101] Léger, J., Brion, E., Desbordes, P., De Vleeschouwer, C., Lee, J. A., Macq, B. (2020). Cross-domain data augmentation for deep learning-based male pelvic organ segmentation in cone beam CT. *Applied Sciences*, 10(3), 1154.

Domain adversarial networks and intensity-based data augmentation for male pelvic organ segmentation in Cone Beam CT

Brion, E., Léger, J., Barragán-Montero, A.M. , Meert, N., Lee, J.A., Macq, B.

Accepted in *Computers in Biology and Medicine*.

Unrelated papers in peer-reviewed journals and conference proceedings.**Modeling patterns of anatomical deformations in prostate patients undergoing radiation therapy with an endorectal balloon**

[19] Brion, E., Richter, C., Macq, B., Stützer, K., Exner, F., Troost, E., Hölscher, T., Bondar, L. (2017, March).

Modeling patterns of anatomical deformations in prostate patients undergoing radiation therapy with an endorectal balloon. In *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling* (Vol. 10135, p. 1013506). International Society for Optics and Photonics.

Acronyms

AI	Artificial intelligence
ANN	Artificial neural network
CBCT	Cone beam computed tomography
CNN	Convolutional neural network
CRF	Conditional random field
CT	Computed tomography
DIR	Deformable image registration
DL	Deep learning
DSC	Dice similarity coefficient
EBRT	External beam radiation therapy
FBA	Federated Byzantine agreement
FCN	Fully convolutional network
FHE	Full homomorphic encryption
GAN	Generative adversarial network
GPU	Graphical processing unit
GRL	Gradient reversal layer
LoA	Limit of agreement
HD	Hausdorff distance

HU	Hounsfield unit
JI	Jaccard index
MR	Magnetic resonance
MRI	Magnetic resonance imaging
OAR	Organ at risk
PCA	Principal component analysis
PCT	Planning CT
PSM	Patient-specific model
ReLU	Rectified linear unit
RT	Radiation therapy
SGD	Stochastic gradient descent
SMBD	Symmetric mean boundary distance
TLS	Transport layer security
UDA	Unsupervised domain adaptation

Contents

Abstract	5
Acknowledgments	7
List of publications	9
Acronyms	11
1 Introduction	17
1.1 Radiotherapy	17
1.2 Challenges	19
1.3 Contributions and outline of this thesis	20
2 Background	23
2.1 Artificial intelligence in medicine	23
2.2 Registration-based segmentation	24
2.3 Patient-specific models	28
2.3.1 Formulating generative modeling as a learning problem	28
2.3.2 From shape generation to image segmentation . . .	30
2.4 Artificial neural networks	31
2.4.1 Formulating image segmentation as a learning problem	31
2.4.2 Artificial neuron	33
2.4.3 Artificial neural network	33
2.4.4 Hyper-parameters and validation	35
2.5 Deep learning	37
2.5.1 Convolutional neural networks	38

2.5.2	Networks with high resolution outputs	45
2.6	Enforcing spatial consistency for segmentation	47
2.6.1	Conditional random fields	47
2.6.2	Adversarial networks	50
2.7	Challenges related to data acquisition and labeling	50
2.7.1	Technical challenges	51
2.7.2	Ethical and legal challenges	53
2.7.3	Inadequate incentives	54
2.8	Domain adaptation	55
3	Secure Architectures Implementing Trusted Coalitions for Blockchain Distributed Learning (TCLearn)	59
3.1	Introduction	61
3.2	Threats and Security goals	63
3.2.1	Threat 1: Keep control over the data	63
3.2.2	Threat 2: Keep control over the model	64
3.3	A scalable security architecture for trusted coalitions	65
3.3.1	Architecture of TCLearn-A	66
3.3.2	Architecture of TCLearn-B	71
3.3.3	Architecture of TCLearn-C	75
3.3.4	Additional features	78
3.4	Security analysis	79
3.4.1	Solution to Threat 1: Keep control over the data	79
3.4.2	Solution to Threat 2: Keep control over the model	81
3.5	Implementation and evaluation	82
3.6	Conclusions	83
4	Contour Propagation in CT Scans with Convolutional Neural Networks	87
4.1	Introduction	88
4.2	Materials and Preprocessing	90
4.3	Formulating the Contour Propagation as a Learning Problem	91
4.3.1	Prior Definition and Computation	92
4.3.2	Network Architecture and Learning Strategy	92
4.4	Results and Discussion	94
4.4.1	Validation Metrics and Comparison Baselines	94
4.4.2	Discussion	97

4.5	Conclusion	100
5	Using planning CTs to enhance CNN-based bladder segmentation on cone beam CT	103
5.1	Introduction	103
5.2	Materials and methods	106
5.2.1	Data and pre-processing	107
5.2.2	Network architecture	107
5.2.3	Learning strategy and performance assessment	108
5.2.4	Comparison baselines	109
5.3	Results and discussion	111
5.4	Conclusion	112
6	Cross-domain data augmentation for deep-learning-based male pelvic organ segmentation in cone beam CT	115
6.1	Introduction	116
6.2	Materials and Methods	119
6.2.1	Data and Preprocessing	119
6.2.2	Model Architecture and Learning Strategy	120
6.2.3	Validation and Comparison Baselines	122
6.3	Results	124
6.4	Discussion	128
6.5	Conclusions	135
7	Domain adversarial networks and intensity-based data augmentation for male pelvic organ segmentation in Cone Beam CT	137
7.1	Introduction	139
7.2	Related works	142
7.2.1	Deep domain adaptation	142
7.2.2	Feature-level transferring	143
7.2.3	Image-level transferring	144
7.2.4	Label-level transferring	145
7.2.5	CBCT segmentation	146
7.3	Materials and methods	146
7.3.1	Data and preprocessing	147
7.3.2	Adversarial networks	147
7.3.3	Intensity-based data augmentation	151

7.3.4	Performance metrics and comparison baselines . .	152
7.4	Results	154
7.4.1	Adversarial networks	154
7.4.2	Intensity-based data augmentation	155
7.5	Discussion	161
7.6	Conclusions	164
8	Conclusion	167

Chapter 1

Introduction

1.1 Radiotherapy

Cancer is a large group of diseases that can start in almost any organ or tissue of the body when abnormal cells grow uncontrollably, go beyond their usual boundaries to invade adjoining parts of the body and/or spread to other organs. It is the second leading cause of death globally, accounting for an estimated 9.6 million deaths in 2018 (one in six deaths), and expected to rise by 35% in 2030 [4]. One of the most widely used treatments is radiotherapy (RT), which should be prescribed to more than half of all cancer patients, either alone or in combination with surgery and/or chemotherapy [38].

In radiotherapy, various types of radiation are used to kill cancer cells. In this thesis, we focus on the most common form of RT called *external beam radiotherapy* (EBRT), in which the patient lies on a couch while ionizing radiations are targeted to the tumor. Most often, the radiation consists of photons, but in some cases, protons are used. Giving an RT treatment involves a trade-off, since delivering the prescribed dose to the target comes at the cost of delivering radiation to healthy tissues (called organs at risk, or OARs) as well. Since delivering overly large doses to OARs leads to undesirable effects and can induce secondary cancer, constraints of dose are imposed on those structures. The workflow presented in Fig. 1.1 is aimed at achieving the prescribed dose to the target while respecting the OAR dose constraints. After a patient has received an indication for radiotherapy, a computed tomography (CT)

scan is acquired for treatment planning. It is therefore called *planning CT*, or PCT. In this image, medical doctors delineate the target volumes as well as the OARs.

For the target, several volumes are often contoured. The gross tumor volume (GTV) is the tumor that is visible on the image. The clinical target volume (CTV) is the GTV plus prior knowledge included by the medical doctor about possible infiltrations of the tumoral cells into surrounding tissues. Also, the planning target volume (PTV) includes margins that account for uncertainties in patient position. Again, this information is not present in the image itself and comes from a priori knowledge and practices that differ across hospitals.

Based on these contours, a physicist proposes a dose plan, i.e., a configuration (such as beam angles and energy levels) of the radiation machine that leads to a dose delivery to the target that is close enough to the prescribed dose, and a dose to the OARs that does not exceed the maximum value allowed by the medical doctor. This involves trade-offs and the final dose plan is the result of several iterations between medical doctor and physicist. The next step is the dose quality assessment, aimed at making sure that the machine delivers the dose distribution that it is supposed to. This is done by positioning a water tank (which simulates a human body, mostly made of water) and measuring the dose with a detector. All these steps (planning CT scan, delineation, dose planning, and quality assessment) constitute the treatment planning and are done once. Then starts the treatment delivery, where the dose is delivered to the patient. The dose is not delivered at once but in several (~ 20) treatment sessions called *fractions*. Each fraction consists of two steps: (i) a new image (called *daily image*) is acquired and used to position the patient at the same place as during planning, and (ii) the dose is delivered. Different image modalities exist for daily imaging, but the one that interests us in this thesis is Cone Beam CT (CBCT). Similar to CT, CBCT is acquired by sending X-rays through the patient's body. While physical constraints prevent the CT scanner and the dose delivery machine from being in the same place, such constraints do not exist for CBCT scanners. However, CBCTs have lower contrast than CT due to additional artifacts such as noise, beam hardening, and scattering.

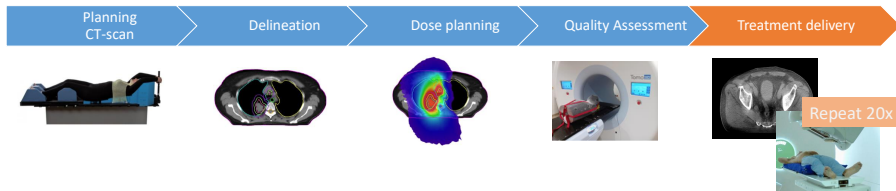


Figure 1.1: The radiotherapy workflow. Adapted from Di Perri and Geets (2015) [39].

1.2 Challenges

While radiotherapy has been used to improve the outcome of patients with cancer for decades, the four following challenges are still open issues for which there is room for treatment improvement.

The contouring step is slow Current practice necessitates annotating the targets and OAR on each slice of the volumetric planning CT. Even though in reality only a few slices are contoured and the slices in-between are interpolated, this step still takes two to four hours, depending on the number of OARs and the difficulty of the case. Slow workflow escalates healthcare costs, reduces doctors' face-to-face time with their patients, and prevents some patients from accessing the treatment that is best suited for them when it is too expensive, such as proton-based radiotherapy.

Inter-fraction variations are poorly monitored The workflow presented in the previous section can be said to be *non-adaptive*, in the sense that the planning phase is done once and does not adapt to anatomical changes (variations in shape, volume, position, and density) occurring during the treatment. To mitigate such inter-fraction variations, some authors have proposed adaptive strategies [6,11,58,118,131]. One such strategy consists of a flagging system looking for deviations between the dose that was intended and the dose delivered at a given fraction. This could be done by comparing the position of the OARs and targets on the planning CT and the daily image. If these positions are similar, the plan is delivered. If the positions are too different, the system asks for a check by a doctor, which can lead to a re-planning

of the patient for the worst deviations (i.e., retake a planning CT scan, redelineate, replan the dose, and eventually redo a quality assessment). A more advanced version of adaptive therapy would not only take as input the position of the organs, but also the dose distribution that is about to be delivered to these structures. The barrier that prevents the application of adaptive radiotherapy today is the workload. Indeed, it requires having contours of the daily image, which are time-consuming to produce manually. This poor monitoring leads to target under-dosage and OAR over-dosage.

It is difficult to gather large, multi-centric datasets in a fast and secure fashion A promising tool for automated planning CT and daily image contouring is deep learning. Deep neural networks are models that learn hierarchical levels of representations of data and have outperformed the state of the art in many applications, including automated contouring [98]. However, these models require large databases of annotated (i.e., already contoured) images. This can be done manually, i.e., by extracting, anonymizing, and centralizing the data. However, manual data collection is at best time-consuming, and at worst impossible when hospitals simply refuse to let data leave their servers.

There could be significant variability between the contours produced by different medical doctors Automated contouring and machine learning open the path to consensus methods for which a solution will be derived by aggregating contours provided by algorithms and several medical doctors.

1.3 Contributions and outline of this thesis

In the previous section, we described four caveats of the current radiotherapy workflow: (i) the slowness of the contouring step in the planning process, (ii) the poor monitoring of inter-fraction variations, (iii) the difficulty of gathering large, multi-centric datasets to train and validate models, and (iv) inter-observer variabilities. In this thesis, we address these challenges with four contributions:

TClean to share knowledge across medical centers We propose a new method based on blockchain, called *Trusted coalition learning*, to collect data for the training and validation of models across different centers in a secure fashion.

Contour propagation Propagation of contours of given structures from manual contours of some slices of the image.

Cross-domain data augmentation to share knowledge between CT and CBCT While most radiotherapy clinics lack annotated CBCTs to train segmentation models, they often have large databases of annotated CTs (since, as we saw in the previous section, contouring them is part of the clinical workflow). We propose a method, called *cross-domain data augmentation*, which leverages large databases of CTs to help train deep neural networks for CBCT contouring, requiring only a limited number of annotated CBCTs in the training set.

Unsupervised domain adaptation to share knowledge between CT and CBCT Here, we propose two strategies that use only the data already available in most radiotherapy clinics: annotated CTs and non-annotated CBCTs. The first method is based on adversarial networks, while the second is based on intensity-based data augmentation.

This document is organized as follows. In Chapter 2 we review the background knowledge upon which our contributions are built. This includes previous work in automatic segmentation for radiotherapy and deep learning. In Chapter 3, we address the problem of data acquisition for medical image analysis and in particular, we compare two approaches. First, we look at manual acquisition, which has most often been done in the past but is showing its limitations with the advent of data-hungry deep learning algorithms. Second, we propose an approach based on blockchain to allow decentralized use of medical data for model training and testing. The third chapter proposes a semi-automatic method for planning CT segmentation, which is shown to be faster than manual contouring and more precise than fully automated methods. In Chapter 5, we propose a way to use planning CTs to improve the segmentation of the bladder, an OAR in prostate cancer, on

Cone Beam CT. These results are extended to the rectum and prostate segmentation in Chapter 6. A limitation of the method proposed in Chapters 5 and 6 (called *cross-domain data augmentation*) is that it still requires manually labeling a few CBCTs to train the model. In Chapter 7 we propose two other methods, one based on adversarial networks and the other on intensity-based data augmentation, to contour the bladder, rectum, and prostate on CBCT without any annotated CBCTs. This document ends with a conclusion about the lessons learned in our work, as well as a perspective on what remains to be done.

These problems were solved using a specific type of widely popular convolutional neural networks known as u-nets. Recently-emerged questions about the explainability of such neural networks and their robustness against adversarial examples in the context of organ segmentation [61, 170] fell outside the scope of this thesis and were left for future work, as the primary focus of this thesis was to improve the quality of radiotherapy treatments in the clinic.

Chapter 2

Background

2.1 Artificial intelligence in medicine

The interest in automatic contouring for radiotherapy belongs to the broader context of artificial intelligence for medicine. Medicine today is confronted with major challenges [158]. As physicians are under pressure to be more productive, the average length of a clinic visit in the U.S. has come down to seven-minute for an established patient; twelve minutes for a new patient. Increased pressure has also led to burnout symptoms in half of the doctors practicing in the U.S. today, and hundreds of suicides each year. This feeds misdiagnosis: 12 million false diagnoses are estimated to happen each year. This in turn is linked to unnecessary medical operations, amounting to one-third of the total [158]. Probably to protect themselves against possible lawsuits, doctors order too many examinations. It is estimated that 30 to 50 percent of the 80 million CT scans in the U.S. are unnecessary. Doctors are also subject to the human limits of biases and slowness. One example of bias is overconfidence. In a study, clinicians who were “completely certain” of the diagnosis antemortem were wrong 40 percent of the time. Humans are also slow. Doctors cannot keep up with the research papers published every day, and a radiologist cannot see in a whole career as many images as a computer in an hour.

Artificial intelligence is likely to be part of the solution. There exist several definitions of artificial intelligence. In this thesis, we will use the following one: *artificial intelligence* (A.I.) is the ability for a non-

human agent to accomplish complex tasks autonomously. We are aware that “complex” is a vague term. Defining it more precisely is subject to philosophical considerations that are beyond the scope of this discussion. Let us just already specify that according to that definition, A.I. does not especially necessitate data, and a computer with non-trivial “if-else” pre-programmed instructions can be considered to display intelligent behavior.

A specific class of A.I. algorithms that has gained interest recently is that of machine learning algorithms. These are algorithms that can learn from data [59]. The concept of *learning* can be defined as follows: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .” Note that machine learning is closely associated with statistics. In their book *Deep Learning* [59], Goodfellow et al. argue that machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to estimate complicated functions statistically and a decreased emphasis on proving confidence intervals around these functions.

In this thesis, we focus on a specific class of machine learning algorithms (deep learning) for a specific task (image segmentation in radiotherapy). In computer vision, *image segmentation* is the attribution of a class to each pixel of an image. But before delving into the specifics of deep learning for image segmentation, we review the main A.I. algorithms for segmentation in radiotherapy (see Venn diagram in Fig. 2.1). In Section 2.2 we present registration-based segmentation, an artificial intelligence algorithm. In Section 2.3 we introduce patient-specific models, an example of a machine learning algorithm. In Section 2.4 we describe the artificial neuron, the building block of an artificial neural network. Finally, we review different deep learning architectures in Section 2.5.

2.2 Registration-based segmentation

Image registration is the task of finding the spatial relationship between two or more images [87]. Once this mapping is known, it can be applied to the known contours of an object in one image to deform them into

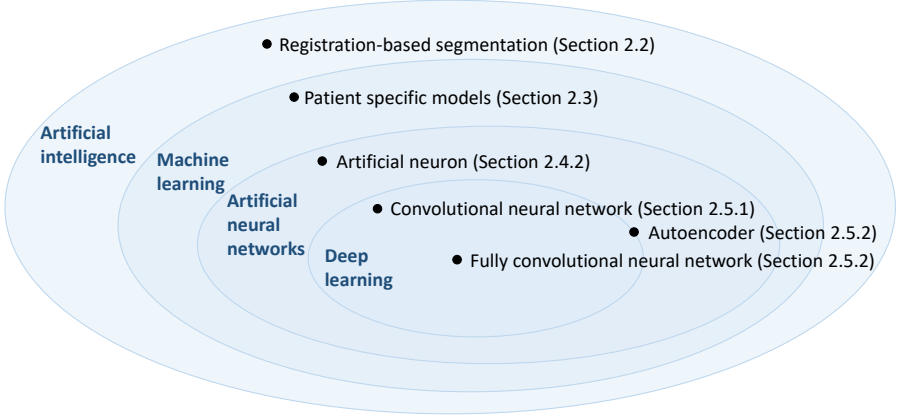


Figure 2.1: Venn diagram of segmentation algorithms used in radiotherapy. Autoencoders belong to the “deep learning” category when they have several “hidden layers”, otherwise they only belong to the “artificial neural network” category.

another, thereby providing a prediction for the contour of the object in the second image (see Fig. 2.2). In the context of radiotherapy, contours correspond to the physical limits of target volumes (i.e., the tumor plus margins) and OARs. The image on which the contours are known is called the *moving image*, while the image on which the contours have to be predicted is the *fixed image* (also called *target image*). In the context of radiotherapy, the fixed image is either the planning CT or the CBCT. In the former case, the moving image is usually the CT of another patient. In the latter case, the moving image is most often the planning CT of the same patient. Formally, images are defined as functions for which the input is a position $\mathbf{x}$ and output the intensity at that position $I_F(\mathbf{x})$ and $I_M(\mathbf{x})$ for the fixed and moving images, respectively. The goal is to find a deformation T that minimizes the error of correspondence between the two images after the deformation. The error is measured with a cost function $\mathcal{C}$:

$$\hat{T} = \arg \min_T \mathcal{C}(T; I_F, I_M), \text{ with} \quad (2.1)$$

$$\mathcal{C}(T; I_F, I_M) = -\mathcal{S}(T; I_F, I_M) + \gamma \mathcal{P}(T). \quad (2.2)$$

In this expression, $\mathcal{S}$ measures fit to data while $\mathcal{P}$ regularizes the deformation field (to be defined). The balance between these two contradictory objectives is controlled by the trade-off parameter γ . When γ is set too low, it can lead to unrealistic transformations. When it is too large, the found transformation may be not accurate enough. Therefore the optimal value for γ must be carefully chosen.

To solve this minimization problem, there are two approaches, parametric and non-parametric. In the following, we discuss both approaches.

Parametric registration In this approach, the deformation T is explicitly described through a model with parameters μ . The problem 2.2 is equivalent to the problem of finding the values of the parameters of the transformation model leading to the best match between fixed and moving images:

$$\hat{\mu} = \arg \min_{\mu} \mathcal{C}(\mu; I_F, I_M). \quad (2.3)$$

An example of a cost function when the fixed and moving images are acquired with different modalities (such as CT and CBCT) is the mutual information:

$$\text{MI}(\mu; I_F, I_M) = \sum_{m \in L_M} \sum_{f \in L_F} p(f, m; \mu) \log_2 \left(\frac{p(f, m; \mu)}{p_F(f)p_M(m; \mu)} \right), \quad (2.4)$$

where L_F and L_M are sets of regularly spaced intensity bin centers, p is the discrete joint probability, and p_F and p_M are the marginal discrete probabilities of the fixed and moving images, obtained by summing p over m and f , respectively. Joint probabilities can be estimated using B-spline Parzen windows (see [86] for more details).

The rigid registration is a simple transformation model:

$$T_{\mu}(\mathbf{x}) = \mathbf{R}(\mathbf{x} - \mathbf{c}) + \mathbf{t} + \mathbf{c}, \quad (2.5)$$

with $\mathbf{R}$ a rotation matrix, $\mathbf{c}$ the center of rotation and $\mathbf{t}$ a translation vector. For a two-dimensional image, the *bending energy penalty* is an example of regularization that penalizes large values of the Hessian of $T(\mathbf{x})$:

$$\mathcal{P}(\mu) = \frac{1}{|\Omega_F|} \sum_{\tilde{\mathbf{x}} \in \Omega_F} \left\| \frac{\partial^2 T}{\partial \mathbf{x} \partial \mathbf{x}^\top}(\tilde{\mathbf{x}}) \right\|_F^2, \quad (2.6)$$

where Ω_F is the domain of the fixed image and $\|\cdot\|_F$ is the Frobenius norm.

The minimization problem is solved iteratively with the gradient descend algorithm. The optimal value is estimated at iteration $k + 1$ from its value at the previous iteration following the formula:

$$\mu_{k+1} = \mu_k - \lambda \left. \frac{\partial \mathcal{C}}{\partial \mu} \right|_{\mu_k}, \quad (2.7)$$

where λ is a parameter of the method called *learning rate*.

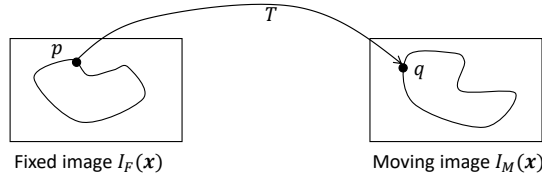


Figure 2.2: Registration.

Non-parametric A limitation of parametric methods is that the regularization field is defined globally. However, the level of regularization that is appropriate often depends on the region. For example, larger deformations are expected to be observed in the bladder (which can experience large differences in filling) than for the femur (bones experience less variability). Therefore non-parametric methods do not have a global model of deformation. Instead, each pixel has its unconstrained deformation vector. This can lead to unrealistic deformations and local regularizations are therefore applied, such as Gaussian smoothing of the deformation vector. Examples of non-parametric methods are the Demons [155] and Morphons [88] algorithms. By default, these algorithms find transformations that are defined in one direction only. For radiotherapy, diffeomorphic¹ versions have also been proposed since they

¹The transformation T is said to be diffeomorphic if it is invertible, differentiable, and its inverse is differentiable [74].

are supposed to be more anatomically realistic [74].

2.3 Patient-specific models

A limitation of registration methods is that they fail when differences between the fixed and the moving images cannot be captured by a deformation model, whether local or global (e.g., when matter appears and disappears). A patient-specific model is a machine learning algorithm that assumes that a few contours are already available for the target patient on images acquired previously. It works in two steps. First, a generative model is learned from the available contours. Second, this model is used to generate the most likely contour for the target image.

2.3.1 Formulating generative modeling as a learning problem

Let us specify each element of a machine learning algorithm (task, experience, and performance) in the specific case of generative modeling.

Task The machine learning task associated with shape generation is called *synthesis* [59]. It consists of generating new samples that are similar to those in the training data.

Experience Machine learning algorithms can be broadly categorized as *unsupervised* or *supervised* by the kind of experience they are allowed to have during the learning process. Unsupervised learning algorithms experience a dataset containing many features, then learn useful properties of the structure of this dataset. Supervised learning algorithms experience a dataset containing features, but each example is also associated with a *label* or a *target*. Let us also mention that some machine learning algorithms do not just experience a fixed dataset. For example, reinforcement learning algorithms interact with an environment, so there is a feedback loop between the learning system and its experiences. Statistical shape models are generative models built in an unsupervised way using a linear dimensionality reduction method called *principal component analysis* (PCA). In these models, the contours are discretized and

represented as point clouds, also called *shapes*. A shape is a concatenation of the 3D geometric coordinates of all points representing the contour. The shapes are represented in a deformation space and then rewritten in a new coordinate system where the first axes represent the directions of main variability among the shapes in the deformation space. By “main direction of variability” we mean the direction that maximizes the variance when the shapes are projected on this axis with PCA (see Fig. 2.3). In this context, it acts as noise filtering: it keeps only the directions of main motion and discards other directions as noise. More specifically, let us suppose that volumetric contours are available for N images of a given target patient. The contours of the i th image can be represented by a vector with 3D geometric coordinates $\mathbf{p}_i \in \mathbb{R}^{3L}$, where L is the number of landmarks in the point cloud (an example with $N = 5$ is depicted in Fig. 2.3a). The mean shape $\bar{\mathbf{p}}$ is the average position of each landmark across all N images:

$$\bar{\mathbf{p}} = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i. \quad (2.8)$$

The covariance matrix is given by

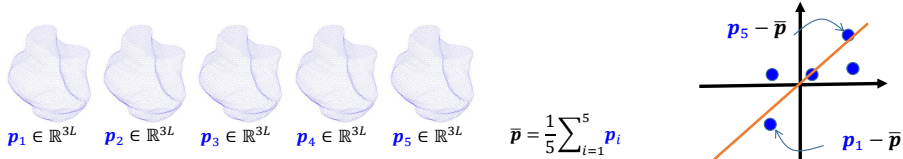
$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^\top. \quad (2.9)$$

Let $\mathbf{q}_j$ be the j th eigenvector of $\mathbf{S}$ and λ_j be the associated eigenvalue (i.e., the variance). Then the i th shape can be approximated by

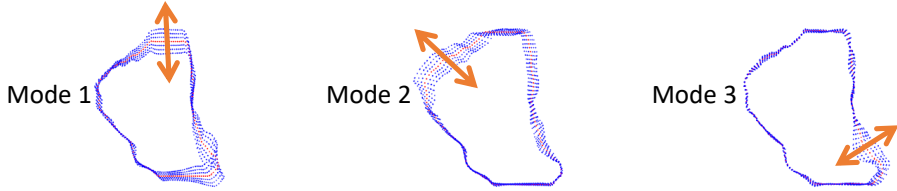
$$\mathbf{p}_i \approx \bar{\mathbf{p}} + \sum_{j=1}^c b_j \mathbf{q}_j, \quad (2.10)$$

where c is a parameter comprised between 1 and a maximum number m which depends on the rank of the matrix $\mathbf{S}$. When c takes its maximum possible value, there is equality between the left- and right-hand sides of the expression. Each element in the sum of Eq. 2.10 corresponds to a main direction of variability (also called *mode*) observed in the training set. Different choices of b_j correspond to different shapes along the different directions of motions observed in the training set (see Fig. 2.3a for an example with $c = 3$). The model presented in Eq. 2.10 is called

a *generative model*, since different values of b_j generate different shapes. In generative modeling, the experience corresponds to the computation of the mean shape, and the eigenvectors and associated eigenvalues of the covariance matrix.



(a) A principal component analysis in the deformation space. The orange line is the main direction of motion. It is thus the first component of the new coordinate system.



(b) Intensity gradient matching along modes. Red corresponds to the mean shape $\bar{\mathbf{p}}$, while blue are shapes with b_j equal to one, two, and three standard deviations $\sqrt{\lambda_j}$.

Figure 2.3: A patient-specific model for the prostate.

Performance One measure of performance of a generative model is the proportion of the variance captured by the model, i.e., the ratio between $\sum_{j=1}^c \lambda_j$ and $\sum_{j=1}^m \lambda_j$. When a model captures a large proportion of the variance with only a few modes, it is more likely to be useful for shape generation.

2.3.2 From shape generation to image segmentation

For image segmentation, values of b_j in Eq. 2.10 are iteratively updated to provide an estimation for the position of a structure of interest on the target image (see Fig. 2.3b). Each landmark is updated perpendicularly to an initial shape until a stopping criterion is met (e.g., the variation compared to the previous iteration is small).

2.4 Artificial neural networks

As we saw in the two previous sections, registration-based segmentation fails when the difference between the source and the target image is too large. This difficulty is partially overcome by patient-specific models, which suffer in turn from another drawback: they require several annotations of images from the target patient, which is time-consuming. To overcome these limitations, we describe in this section how image segmentation can be framed as a learning problem. Then we define a building block, the artificial neuron, before showing how it can be assembled into more complex architectures called artificial neural networks.

2.4.1 Formulating image segmentation as a learning problem

Image segmentation can be formulated in the context of machine learning, with a specific task, experience, and performance.

Task The machine learning task associated with image segmentation is *classification*. In this type of task, the computer program is asked to specify which of k categories some input belongs to. To solve this task, the learning algorithm is usually designed to produce a function $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$. When $y = f(\mathbf{x})$, the model assigns an input described by a vector $\mathbf{x}$ to a category identified by numeric code y . This general framework for image classification needs adaptations for segmenting 3D images. Three main strategies have been proposed to formulate the problem of segmenting an image of size $w \times l \times h$ as a classification task: patch-based, tri-planar, and end-to-end.

In the patch-based strategy [23], the input is a patch $\mathbf{x} \in \mathbb{R}^{p_1 \times p_2}$ chosen in a slice of the volumetric image and the output is the category of the central pixel of this patch ($y \in \{1, \dots, k\}$). Alternatively, patches can be 3-dimensional. In segmentation, $k - 1$ categories correspond to structures of interest (such as organs) supposed to be present in the image while there is an additional category accounting for the background. Each voxel in the volumetric image is classified by extracting a different $p_1 \times p_2$ patch in its neighborhood. A limitation of this method is that information is considered in a given slice only. Relevant information about

intensities in the slices above and below the voxel of interest are discarded. Therefore, an alternative strategy does not take only one patch in a given slice but rather takes three patches belonging to intersecting planes ($\mathbf{x} \in \mathbb{R}^{p_1 \times p_2 \times 3}$). It can be called the “tri-planar” strategy [139] and in this case, the machine learning algorithm predicts the category of the voxel at the intersection of the three patches ($y \in \{1, \dots, k\}$). Here again, for each voxel to be segmented, three different patches are extracted. This strategy is still limited since some voxels in the vicinity of the voxel of interest are not considered for predicting its category. When classifying the voxel of coordinates (x, y, z) , for instance, the voxel of coordinates $(x + 1, y + 1, z + 1)$ is in its vicinity yet it does not belong to any of the three planes. To mitigate this, the authors in [120, 143] take as input the whole volume ($\mathbf{x} \in \mathbb{R}^{l \times w \times h}$) and predict the category of all voxels in this volume ($\mathbf{y} \in \{1, \dots, k\}^{l \times w \times h}$). This end-to-end strategy has the drawback of requiring large volumes to be loaded in the GPU memory.

Experience In the context of medical image segmentation, supervised learning is the most common setting and the labels correspond to the class of each voxel. In supervised learning, training often takes the form of the minimization of a loss function between the target and a model prediction with an optimization algorithm.

Performance The performance of segmentation algorithms can be measured with overlap-based and distance-based metrics. The Dice similarity coefficient (DSC) and Jaccard index (JI) are common overlap-based metrics, while the symmetric mean boundary distance (SMBD) is a common distance-based metric:

$$\text{DSC} = \frac{2|M \cap P|}{|M| + |P|}, \quad (2.11)$$

$$\text{JI} = \frac{|M \cap P|}{|M \cup P|}, \quad (2.12)$$

$$\text{SMBD} = \frac{\overline{D}(M, P) + \overline{D}(P, M)}{2}, \quad (2.13)$$

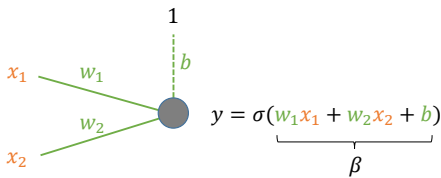
where M and P are the sets containing the matricial indices of the manual and predicted segmentation 3D binary masks, respectively, $\overline{D}(M, P)$ is the mean of $D(M, P)$ over the voxels of Ω_M , and $D(M, P) = \{\min_{x \in \Omega_P} \|\mathbf{s} \odot (\mathbf{x} - \mathbf{y})\|, \mathbf{y} \in \Omega_M\}$, where Ω_M and Ω_P are the boundaries extracted from M and P , respectively, and $\mathbf{s}$ is the pixel spacing in mm. In this expression, $\odot$ is the elementwise product between vectors.

2.4.2 Artificial neuron

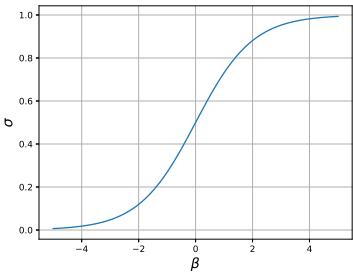
We mentioned above that machine learning algorithms for classification are asked to produce a function $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$. We now specify the forms that this function f can take. A simple example is the *artificial neuron*. The artificial neuron takes as input a vector $\mathbf{x} \in \mathbb{R}^n$ (an example with $n = 2$ is illustrated at 2.4a) and outputs a scalar $y = g(\mathbf{w}^\top \mathbf{x} + b)$, where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the model parameters and g is a non-linear function called *activation*. The presence of non-linearities is one of the key differences compared to patient-specific models. The latter are based on principal component analysis, which works with matrix algebra and therefore is only linear. A popular choice for g is the sigmoid function, defined as $\sigma(z) = 1/(1 + \exp(-z))$ (see 2.4b). Another choice is $g(z) = \max(0, z)$, which is called the *rectified linear unit* or ReLU. For the segmentation of a single structure of interest ($k = 2$, the first category corresponding for example to “lung” and the other to “non-lung”) with the patch-based strategy, the input contains the patch intensities and the output is the probability of the central voxel of this patch’s belonging to the lung. If $y > 0.5$, the voxel is predicted to belong to the lung, otherwise it is predicted to be a non-lung voxel (the sigmoid has the property of taking values in the range $[0, 1]$ only).

2.4.3 Artificial neural network

While the artificial neuron provides a useful class of models for many machine learning applications, it lacks flexibility for image segmentation. Indeed, most correspondences between a patch and the probability of its central pixel belonging to, say, the lung cannot be properly modeled by an artificial neuron. Another limitation is that an artificial neuron cannot predict among more than $k = 2$ categories. A way to overcome these two limitations is to stack several artificial neurons together to



(a) An artificial neuron.



(b) The sigmoid function
 $\sigma(\beta) = 1/(1 + \exp(-\beta))$.

Figure 2.4: The artificial neuron and the sigmoid function are common building blocks of deep learning models.

form an *artificial neural network*. This architecture is organized into three types of layers: input layers, hidden layers, and output layers. For image segmentation, input layers simply correspond to the intensities of the image to be segmented. Figure 2.4.3 shows the computation of a single hidden layer with three hidden units. This allows extending the class of models compared to a single neuron. To allow multi-class classification, the output layer has several artificial neurons (one per category). For their activation, the sigmoid is replaced with another activation function called *softmax* and defined by

$$\sigma_j(z_j) = \frac{\exp(z_j)}{\sum_{i=1}^k \exp(z_i)}, \quad (2.14)$$

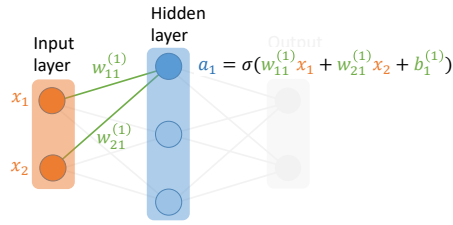
where $z_i = w_{1i}^{(2)} a_1 + w_{2i}^{(2)} a_2 + w_{3i}^{(2)} a_3 + b_i^{(2)}$ for $i = 1, \dots, k$. An artificial neural network (ANN) with a single hidden layer has been shown to be a universal approximator [35, 69]. This means that all functions can be approximated by this model, provided that the hidden layer has enough neurons. However, the number of neurons needed to model a given function with a single hidden layer is often unfeasibly high and the model may not generalize well. For a given number of neurons, arranging them in several hidden layers instead of a single layer is often more practical and leads to better generalization. The study of such artificial neural networks with several hidden layers is called *deep learning*. More generally, let us consider a deep neural network with L hidden layers and an activation function $g^{[l]}$ at layer l . The network's output is governed by the following equations:

$$\begin{aligned} \mathbf{z}^{[l]} &= \mathbf{W}^{[l-1]} \mathbf{a}^{[l-1]} + \mathbf{b}^{[l-1]}, \\ \mathbf{a}^{[l]} &= g^{[l]}(\mathbf{z}^{[l]}), \end{aligned} \quad (2.15)$$

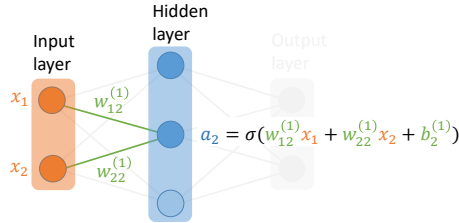
for $1 \leq l \leq L$. In these expressions $\mathbf{a}^{[0]} = \mathbf{x}$ is the network's input, while $\mathbf{a}^{[L]} = \hat{\mathbf{y}}$ is the network's output.

2.4.4 Hyper-parameters and validation

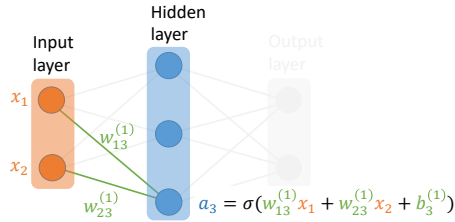
A neural network defines a function $f(\mathbf{x}, \theta)$, where $\mathbf{x}$ is the input and θ is a set of parameters (i.e., a vectors containing the weights and the



(a) First hidden neuron.



(b) Second hidden neuron.



(c) Third hidden neuron.

Figure 2.5: In an artificial neural network, artificial neurons are stacked into hidden layers (in this case a single hidden layer) to extend the class of functions available (compared to an artificial neuron alone).

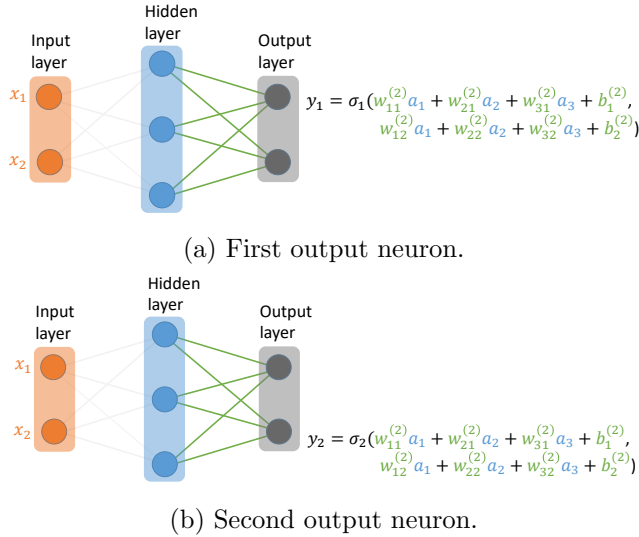


Figure 2.6: Output layer of an artificial neural network.

biases of all layers). These weights are chosen with stochastic gradient descent (SGD), an algorithm based on the derivatives of f with respect to θ . However, other parameters must be fixed, such as the parameters of SGD itself (e.g., learning rate), and the number of artificial neurons per layer. Such parameters that cannot be chosen directly by SGD are called *hyper-parameters*. Stochastic gradient descent chooses biases and weights that maximizes the proximity between $f(\mathbf{x}, \theta)$ and the true label of $\mathbf{x}$ for different values of $\mathbf{x}$ that belong to a set called the *training set*. However, good accuracy on the training set is not enough; performing well on new, unseen data is the ultimate goal. In order to solve these two issues (making sure SGD does not over specialize on the training set and choosing the hyper-parameters), the performance of trained models with different hyper-parameters is evaluated on another set, called the *validation set*.

2.5 Deep learning

In this section, we describe deep architectures used for image classification and segmentation.

2.5.1 Convolutional neural networks

For the networks presented in the previous section, each neuron of a given layer is connected to *all* neurons of the following layer. Although this is useful in many applications, this setting is not the most appropriate when inputs are images. First, the number of connections between a 2D input image and the first hidden layer grows quadratically with the size of the input patch (this is illustrated in Fig. 2.7a). If the input image is volumetric (as can be done for CT scans), the growth is cubic. Large input images are often needed to have sufficient contextual information, and larger networks are harder to train. Second, it lacks a desirable property for classification networks: spatial invariance. To take an intuitive example in natural image classification, a desirable property would be for the network to show similar behavior after a small translation of an object in an image, since it is not supposed to impact the outcome of the prediction (see Fig. 2.7b). For a fully-connected network, small translations in the input image lead to different outputs of the artificial neurons.

The convolutional layer To address these issues, the convolutional layer has been proposed [99] to impose a constraint on the architecture motivated by the domain of application (computer vision). To compute the activation of a hidden neuron, an element-wise multiplication and sum is computed with a given set of weights (this set of weights is called a *filter*, see Fig. 2.8a). The weighted sum is computed with respect to a limited region of the input image only (contrary to fully-connected networks). For the next hidden neuron, the same weights are used but this time the weighted sum is computed according to a different area of the input image. This process is repeated until the whole image is covered. In practice, several sets of weights are used, but the principle of looking at different areas of the input image remains the same. Hidden weights corresponding to the same filter are said to belong to the same *feature map*. This sliding window approach corresponds to a mathematical operation called *convolution*. This leads each filter to act like a pattern detector. More generally, for n_C filters of size $n_H \times n_W$, the k th feature map is computed as

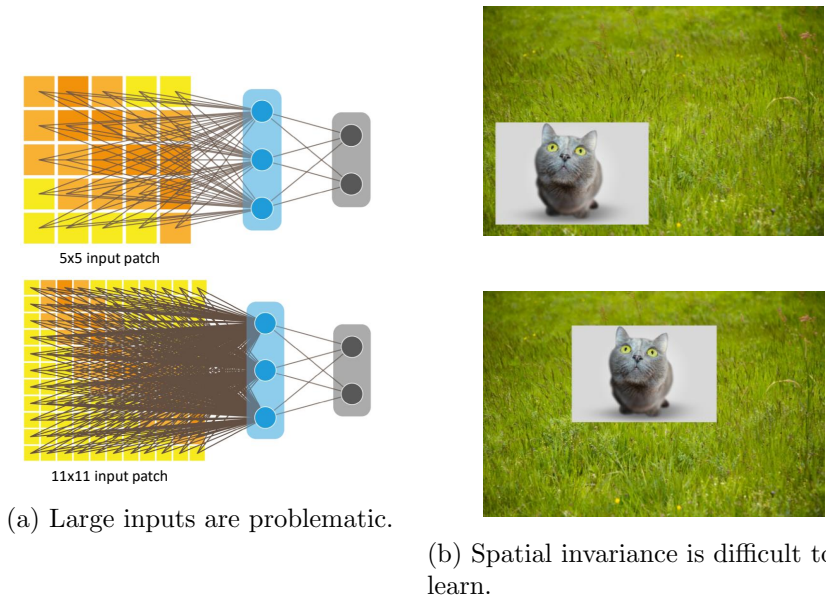
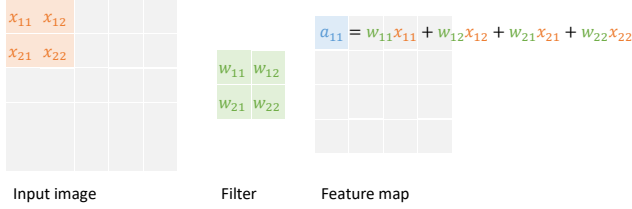
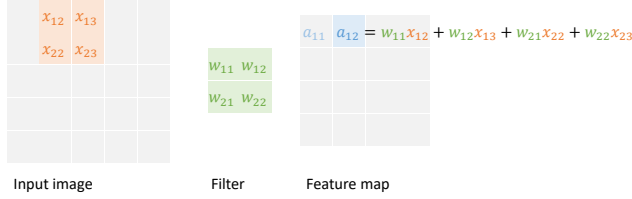


Figure 2.7: Fully-connected artificial neural networks are impractical for images. Source: “Deep learning in medical imaging” workshop at SPIE medical imaging (2017, unpublished).



(a) Computation of the first neuron of a feature map.



(b) Computation of the second neuron of a feature map.

Figure 2.8: A convolutional layer.

$$z_{i,j,k}^{[l]} = \sum_{\alpha=1}^{n_H} \sum_{\beta=1}^{n_W} \sum_{\gamma=1}^{n_C} w_{i,j,k} a_{i+\alpha,j+\beta,\gamma}^{[l-1]} + b_{\gamma}^{[l-1]}, \quad (2.16)$$

$$a_{i,j,k}^{[l]} = g^{[l]}(z_{i,j,k}^{[l]}),$$

where $1 \leq l \leq L$, and for the l th layer $1 \leq i \leq n_x^{[l]}$ and $1 \leq j \leq n_y^{[l]}$, with $n_x^{[l]} = n_x^{[l-1]} - n_H + 1$ and $n_y^{[l]} = n_y^{[l-1]} - n_W + 1$. In particular, $n_x[0]$ is the length of the input image $\mathbf{x}$ in the x dimension, while $n_y[0]$ is the length in the y dimension.

Sometimes, convolution with *padding* is used. This means that zero entries are added around feature maps before convolution. The purpose of padding is to control the resolution of the feature map. Additional variants include convolutions with different stride and dilation rates (see [44] for more details). Networks with one or more such convolutional layers are called *convolutional neural networks* (CNN) or *convNets*.

The max-pooling layer While convolutional layers use efficient weight sharing to look for patterns in images, they do not exhibit the spatial

invariance property. This is illustrated in Fig. 2.9. Let us suppose that the presence or absence of a horizontal bar in the lower left corner of an image has high predictive value for classifying this image. A relevant filter for detecting this pattern is a 2×2 filter with one line where all entries are equal to one and the other entries equal to zero. In the figure, four different images with small translations of a vertical edge are shown. In most applications, such small translations of a pattern in the input image should not impact the outcome of the prediction. In this example, we see that each input image leads to a different feature map after the convolution operation. To introduce translation invariance, a max-pooling operation is often added. This operation consists in replacing neurons in a feature map by their maximum value in a given region of this feature map (in this figure, a 2×2 window). We can see in the figure that each of the four input images lead to the same neuron (lower left) to be activated in the max-pooling feature map. This introduces spatial invariance and lowers the number of weights, thereby reducing overfitting and improving generalization. More generally, the maximum can be taken in any neighborhood of size $n_H \times n_W$ of the k th feature map:

$$a_{i,j,k}^{[l]} = \max_{\substack{1 \leq \alpha \leq n_H \\ 1 \leq \beta \leq n_W}} a_{i-1+\alpha, j-1+\beta, k}^{[l-1]}, \quad (2.17)$$

where $1 \leq l \leq L$, and for the l th layer $1 \leq i \leq n_x^{[l]}$ and $1 \leq j \leq n_y^{[l]}$, with $n_x^{[l]} = n_x^{[l-1]}/2$, and $n_y^{[l]} = n_y^{[l-1]}/2$. Note that there exist alternative ways to reduce the resolution, such as the dilated convolution (which we do not discuss here).

Stacking convolutions, max-pooling and fully-connected layers

Predicting the class of an image most often requires the detection of structures larger than 2×2 pixels. In principle, structures of any size can be detected provided that the filters are large enough. Another possibility to detect large structures is to stack several layers of convolutions and max-pooling with smaller filters. It is such a stacking of convolutions and max-pooling that allowed Krizhevsky et al. to obtain spectacular results of a CNN for image classification, with 11×11 , 5×5 and 3×3 filters [92]. In this architecture, first convolutions detect small patterns, such as edges

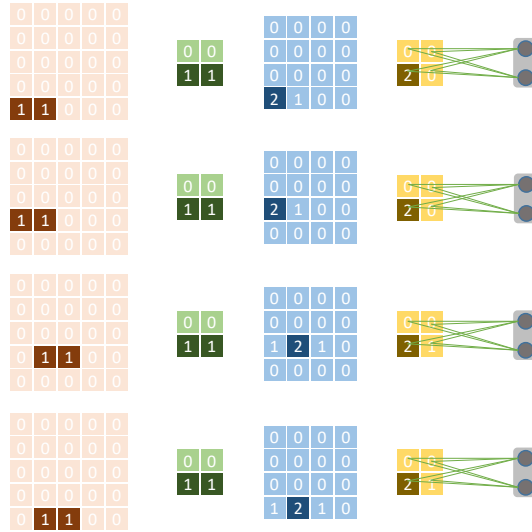


Figure 2.9: An artificial neural network with an input image (red), convolutional layer (filter in green and associated feature map in blue), a max-pooling layer (yellow), and fully-connected layer (grey). In this figure, $n_H = n_W = 2$ for both the convolution and the max-pooling layers.

(see Fig. 2.10). Intermediate layers build upon these simple patterns to look for more complex patterns, such as complicated curves. Deeper convolutions assemble those curves to look for even more abstract objects, such as wheels, heads, or tree branches. Finally, fully-connected layers are used to establish correspondence between different combinations of presence/absence of objects and the general category to which the input image is predicted to belong. The authors in [149] showed that the results improved when more (16-19) layers with smaller (3x3) filters were used, compared with shallower networks with larger filters. Networks of such architecture are called VGG networks, named after the Visual Geometry Group to which the authors belonged.

Residual connections A neural network can be described as a function $f(\mathbf{x}; \theta)$ with input $\mathbf{x}$ and parameters θ , which correspond to the weights and to the biases. A given neural network architecture defines a

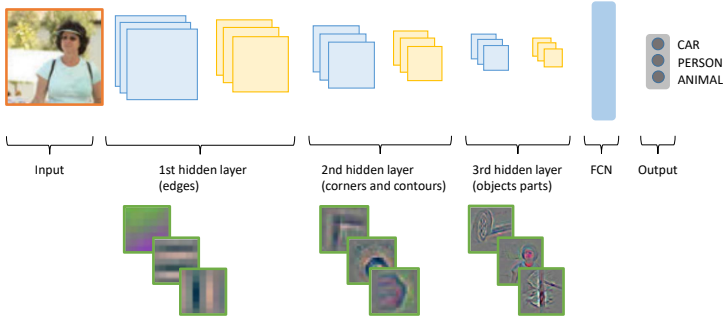
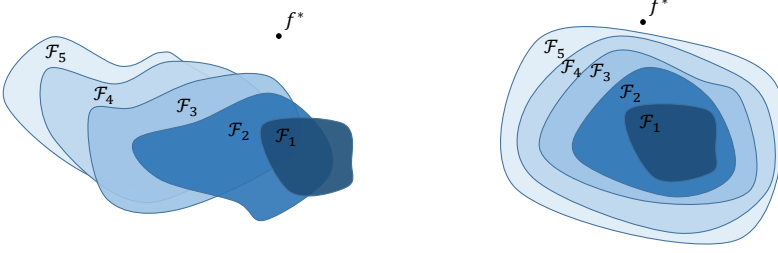


Figure 2.10: Deep neural networks learn through different levels of abstraction. FCL: fully-connected layer. Adapted from Goodfellow et al. (2016) [59].

class of function $\mathcal{F}$: it is the set of functions $\{\mathbf{x} \mapsto f(\mathbf{x}, \theta), \theta \in \Theta\}$ where Θ is the set of parameters θ that can be found with various choices of optimizers and hyper-parameters. Let us suppose a VGG neural network architecture that defines a class of functions $\mathcal{F}_1$. When an additional layer is added, it defines a new class of functions $\mathcal{F}_2$. Since it has more weights, the class of functions $\mathcal{F}_2$ is likely to be broader than $\mathcal{F}_1$. However, some functions in $\mathcal{F}_1$, may not belong to $\mathcal{F}_2$. Indeed, deeper networks are harder to train, partly because they have been shown experimentally to lead to either very large or very low gradients, a common issue known as *the vanishing/exploding gradient problem*. Even though proper initialization partially mitigates this difficulty, it is often not sufficient. Adding layers to $\mathcal{F}_2$ defines even broader classes of function $\mathcal{F}_3$, $\mathcal{F}_4$, and $\mathcal{F}_5$, with $\mathcal{F}_1 \not\subseteq \mathcal{F}_2 \not\subseteq \mathcal{F}_3 \dots$. We say that they are *non-nested* function classes [175]. Now let f^* be the optimal classifier. What is observed in practice is that adding layers allows the network to learn functions closer to f^* *up to a point*. Despite defining broader classes of functions, a VGG network with too many layers typically contains less desirable functions. To overcome this difficulty, He et al. [66] have proposed the *residual layer*. When a residual layer is added to an architecture, it defines a broader class of functions $\mathcal{F}_2$ that includes the original class $\mathcal{F}_1$. This layer is defined as follows:



(a) Non-nested classes of functions. (b) Nested classes of functions.

Figure 2.11: When they define nested classes of functions $\{\mathcal{F}_i\}_{1 \leq i \leq n}$ ($n = 5$ on this illustration), deeper neural networks are more likely to yield close approximations of the optimal classifier f^* . Adapted from Zhang et al. (2019) [175].

$$z_{i,j,k}^{[l]} = \sum_{\alpha=1}^{n_H} \sum_{\beta=1}^{n_W} \sum_{\gamma=1}^{n_C} w_{i,j,k} a_{i+\alpha,j+\beta,\gamma}^{[l-1]} + b_{\gamma}^{[l-1]}, \quad (2.18)$$

$$a_{i,j,k}^{[l]} = g^{[l]}(z_{i,j,k}^{[l]}) + a_{i,j,k}^{[l-1]},$$

where $1 \leq l \leq L$, and for the l th layer $1 \leq i \leq n_x^{[l]}$ and $1 \leq j \leq n_y^{[l]}$, with $n_x^{[l]} = n_x^{[l-1]} - n_H + 1$ and $n_y^{[l]} = n_y^{[l-1]} - n_W + 1$. If the activation function at the l th layer $g^{[l]}$ is the ReLU (as is often the case), then the network can easily choose weights $\mathbf{W}$ and biases $\mathbf{b}$ leading the ReLU to output a zero value, and thus the network with the additional layer can easily produce the same output as the original network. Therefore, adding additional layers defines *nested classes of functions* (i.e., $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \mathcal{F}_3 \dots$). This ensures that adding layers at least does not hurt the network's performance.

Dense connection Another option, *dense connections*, consists in concatenating feature maps at different levels. Let us define a more compact expression for the convolutional layer: $\mathbf{a}^{[l]} = H^{[l-1]}(\mathbf{a}^{[l-1]})$, where $H^{[l-1]}$ corresponds to the two operations of Eq. 2.16. Then the residual connection of Eq. 2.18 can be rewritten as $\mathbf{a}^{[l]} = H^{[l-1]}(\mathbf{a}^{[l-1]}) + \mathbf{a}^{[l-1]}$. The *dense connection*, in contrast, corresponds to

$$\mathbf{a}^{[l]} = H^{[l-1]}([\mathbf{a}^{[0]}, \mathbf{a}^{[1]}, \dots, \mathbf{a}^{[l-1]}]), \quad (2.19)$$

for $1 \leq l \leq L$. In this expression, $[\cdot, \cdot]$ designates the concatenation along the last dimension (i.e., it is a concatenation of the feature maps). Similar to their residual counterpart, dense connections have been shown to alleviate the gradient vanishing/exploding problem while using fewer parameters.

2.5.2 Networks with high resolution outputs

The fully-connected layer, convolution operation, max-pooling, and activation functions presented in the previous subsection are the main building blocks of most image classification networks, where a single class must be predicted for the whole image. In image segmentation, however, each pixel of the image has to be allocated to one of several classes. Patches can then be extracted around each voxel and run separately in the same network to classify the central pixel of the patch [32]. However, this solution is slow, since the network must be run separately for each patch, and there may be redundancy between patches. An obvious solution to these difficulties consists in training from the whole image as input and outputting the segmentation as a whole. In other words, a solution consists in enlarging the output's resolution.

Autoencoders While neural networks with high resolution output have already been proposed long ago, they have been primarily used not for segmentation, but for non-linear dimensionality reduction. The neural network used for this task is a particular case of the ANN called *autoencoder*, where input and output have the same size and each hidden layer usually has a smaller size than the input size. A simple autoencoder with a single hidden layer is shown in Fig. 2.12. An autoencoder is a neural network that is trained to attempt to copy its input to its output [59]. Internally, it has a hidden layer that describes a code used to represent the input. The network consists of two parts: an encoder and a decoder. If the autoencoder succeeds in copying the input exactly, it is not useful. Therefore, its capacity is limited to force the network to learn useful properties of the input. Even though autoencoders have

been classically used for dimensionality reduction and feature learning, they have recently gained popularity for generative modeling, in which case they are called *variational autoencoders* [40]. Unlike for segmentation, autoencoders are aimed at producing an output that is as close as possible to the input. Another difference is the way the network is trained. Autoencoders are the only types of architecture presented in this chapter that are trained in an unsupervised way (i.e., without labels for the training set).

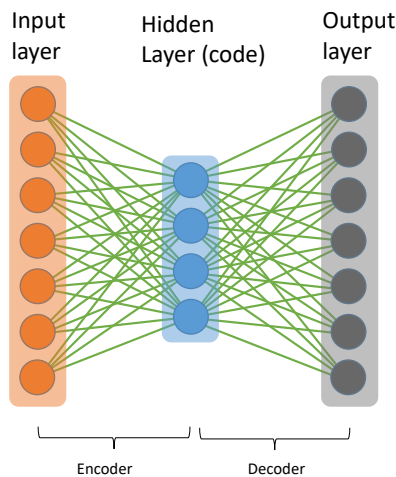


Figure 2.12: A simple autoencoder.

Fully-convolutional neural networks An interesting characteristic of autoencoders is that they learn key features of the input in hidden layers of smaller dimension, and then assemble these features to produce a high-dimensional output. Another kind of architecture called fully-convolutional neural networks (FCN) [109] translates this principle to segmentation tasks. FCNs have two parts. First, the contracting part is similar to CNNs used for classification and decreases the resolution of the input image. Second, the expanding part recovers the original resolution by applying an operation called *transpose convolutions*. Transpose convolutions are defined by the following equations:

$$\begin{aligned}
s_{i,j,k}^{[l-1]} &= \begin{cases} a_{i,j,k}^{[l-1]}, & \text{if } i \text{ and } j \text{ are odd} \\ 0, & \text{otherwise} \end{cases} \\
z_{i,j,k}^{[l]} &= \sum_{\alpha=1}^{n_H} \sum_{\beta=1}^{n_W} \sum_{\gamma=1}^{n_C} w_{i,j,k} s_{i+\alpha,j+\beta,\gamma}^{[l-1]} + b_{\gamma}^{[l-1]}, \\
a_{i,j,k}^{[l]} &= g^{[l]}(z_{i,j,k}).
\end{aligned} \tag{2.20}$$

In FCNs, the contracting path detects the presence of a certain object (such as the bladder), while the expanding path recovers the exact – at the pixel level – delineation of this object. A limitation of classical FCNs, however, is that in the expanding path, precise information about localization has been lost due to successive downsamplings. For this reason, copying feature maps at each resolution from the contracting path to the expanding path has been proposed in a network called “u-net” [143] (see Fig. 2.13). While it was originally proposed for segmentation, it has since been successfully applied to other applications where the input and output have similar resolutions. Examples in medical imaging for RT include the denoising of Monte-Carlo dose simulations for proton therapy [75] and dose prediction for photons RT [10].

2.6 Enforcing spatial consistency for segmentation

In segmentation models such as u-net, the class of each voxel is predicted independently. Even though neighboring voxels share substantial contextual information, local minima of the loss function and noise in the input images can generate spatial inconsistencies in the final segmentation map [81, 91]. Conditional random fields and semantic segmentation using adversarial networks are two methods that mitigate this issue by enforcing spatial continuity.

2.6.1 Conditional random fields

One way to enforce spatial continuity is to encourage voxels close to each other and with similar intensity to belong to the same class. To this end,

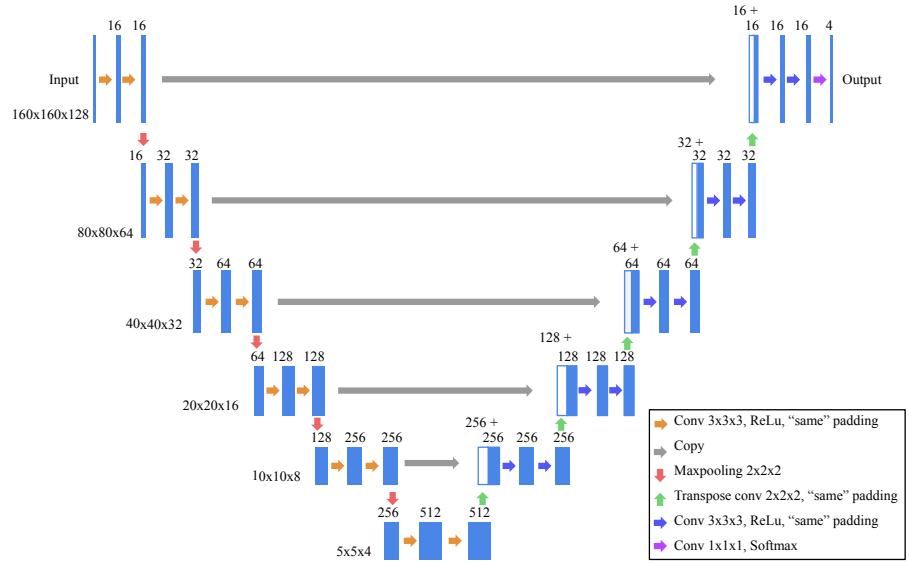


Figure 2.13: A u-net architecture with six-level depth. In this figure, each blue rectangle represents the result of a convolution operation, with the number of feature maps written above it. White rectangles represent feature maps copied from another layer before being concatenated (along the last dimension) to feature maps of another layer (the number of feature maps of this concatenation is the sum of the number of feature maps of the two assembled layers, therefore the “+” sign in those cases). In this particular architecture, “same padding” is used. This means that zero entries are added around the input or the feature maps before convolution. The purpose of padding is to keep the resolution unchanged after convolutions, which allows some feature maps to be concatenated.

a set of labels $\mathbf{z}$ for an entire volumetric image can be associated with a quantity called the Gibbs energy [91]:

$$E(\mathbf{z}) = \sum_i \psi_u(z_i) + \sum_{i \neq j} \psi_p(z_i, z_j). \quad (2.21)$$

In this expression, the first term is the unary potential associated with the label z_i of the i th voxel. It is computed based on the network's output: $\psi_u(z_i) = -\log P(z_i|\mathbf{I})$, where $\mathbf{I}$ is the matricial representation of the entire volumetric input image. The second term is the pairwise potential between any pair of voxels i and j and can be modeled as $\psi_p(z_i, z_j) = \mu(z_i, z_j)k(\mathbf{f}_i, \mathbf{f}_j)$, with $\mathbf{f}_i$ and $\mathbf{f}_j$ the representations of the i th and j th voxels in a feature space. The purpose of the first coefficient is to have a nonzero potential only for different labels. It is called the Pott's model and writes $\mu(z_i, z_j) = [z_i \neq z_j]$. The second factor defines a potential over pairs of feature vectors (for example, activations in feature maps) of voxels i and j . This penalty function can be modeled as a linear combination of Gaussian kernels:

$$k(\mathbf{f}_i, \mathbf{f}_j) = \sum_{m=1}^M w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j). \quad (2.22)$$

In most medical image applications, two Gaussian kernels are used. The first one is the *smoothness kernel* and encourages the attribution of similar labels to voxels that are spatially close:

$$k^{(1)}(\mathbf{f}_i, \mathbf{f}_j) = \exp \left(- \sum_{d=\{x,y,z\}} \frac{(p_{i,d} - p_{j,d})^2}{2\sigma_{\alpha,d}^2} \right), \quad (2.23)$$

where $p_{i,d}$ and $p_{j,d}$ are the coordinates of the i th and j th voxel in the d th dimension (respectively). The second kernel is the *appearance kernel* and encourage to have similar labels for voxels that both are close and have similar intensities:

$$k^{(2)}(\mathbf{f}_i, \mathbf{f}_j) = \exp \left(- \sum_{d=\{x,y,z\}} \frac{(p_{i,d} - p_{j,d})^2}{2\sigma_{\alpha,d}^2} - \sum_{c=1}^C \frac{(I_{i,c} - I_{j,c})^2}{2\sigma_{\gamma,c}^2} \right), \quad (2.24)$$

where C is the number of channels of the input images. The best labeling $\mathbf{z}^* = \arg \min_{\mathbf{z} \in \mathcal{L}^n} E(\mathbf{z})$, with $\mathcal{L}^n$ the set of possible labels when there are n classes, is estimated using the mean field approximation algorithm [91]. The weights and kernels are hyper-parameters. Conditional random fields have been shown to improve segmentation accuracy in multiple applications, including brain lesion [81], liver and tumor [30], and mass [182] segmentation.

2.6.2 Adversarial networks

A limitation of CRFs is that they are still a rigid second-order model. Adversarial networks, thanks to their higher capacity, allow enforcing higher-order interactions. Originally, adversarial networks were developed for generative modeling, i.e. to model the distribution from a given set of images, and then to draw new samples from this distribution [60]. Luc et al. [111] extended this framework for segmentation. In their work, a second network called *discriminator* is added to the segmenter (which can be a u-net, for example). The discriminator takes as input both an image and a segmentation that is either manual or predicted by the segmenter and must classify whether they are human or predicted segmentations. The general framework is shown in Fig. 2.14. The loss function is designed such that the discriminator has an incentive to have high classification accuracy, while the segmenter has (i) high segmentation accuracy and (ii) induces low classification accuracy for the discriminator. In this manner, the segmenter is encouraged to produce segmentations that are globally close to human ones. The segmenter and discriminator have antagonist objectives, therefore the name *adversarial networks*. This method has been successfully used to segment anatomical structures such as the liver [171] or different regions in prostate cancer [89].

2.7 Challenges related to data acquisition and labeling

As mentioned in the previous sections, training and validating deep neural networks requires large datasets. This is currently difficult due to technical, legal, and ethical challenges, and an inadequate system of

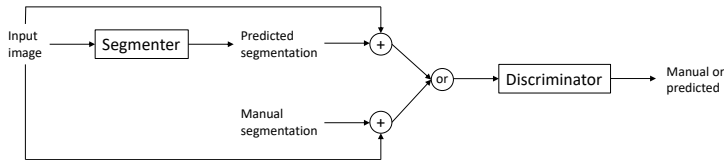


Figure 2.14: Basic framework for image segmentation with adversarial networks. The “+” sign represents the concatenation operation.

incentives.

2.7.1 Technical challenges

Collecting image data involves technical challenges. Once these images are acquired, there are additional challenges to obtain proper labels for that data. In this section, we describe both difficulties.

Images For a given application, the theoretical limit on the number of medical images available is below what is needed to match human performance in most natural image applications. Deep learning has shown its most impressive results on natural images. To match or exceed human performance, the training dataset should contain at least ten million examples [59]. This is feasible for natural images since taking pictures of the outside world is easy and there are already millions of open-source images on the Internet. In contrast, one cannot take a CT scan of a prostate cancer patient with a smartphone and post it online. It would even reduce the dimensionality of the dataset from 3D to 2D and lose resolution. Therefore achieving a database of this size is out of reach for most applications in medical image analysis. In this thesis, for example, we focus on OARs and target segmentation for prostate cancer. Each year, there are 1.28 million new cases of prostate cancer worldwide. CT scans of patients with other pathologies also could have been used to train a model for segmenting OARs for prostate cancer. Yet for target delineation, the amount of training data is limited by the number of patients with prostate cancer. This means that if we started to build today a database containing prostate cancer images of all patients *worldwide*, it would take more than seven years to achieve the size typically required to reach human performance, which is not

practical. Prostate cancer is one of the cancers with the highest incidence (the fourth, men and women combined), which means that building a dataset for other cancers (such as of the stomach, liver, and esophagus) would be even more challenging. Fortunately, although a ten-million-image database is often necessary in natural images to match human performance, a smaller amount than that has been shown to be sufficient for certain applications in medical image analysis: Esteva et al. [49] have reached dermatologist-level performance for classification of skin cancer with 129,450 images, and Rajpurker et al. [140] attained radiologist-level pneumonia detection on chest X-rays using just 112,120 images. Still, gathering on the order of 100,000 images remains challenging, and studies with this order of magnitude of data are still the exception at the time of writing this thesis.

Many images are not stored. After acquisition, images are kept in the database for a limited amount of time (e.g., 5 years), and then deleted. This limits the number of available images.

In a given hospital, it is time consuming to retrieve images. The images can often be retrieved only using a given software. This software requires an expensive license and is therefore installed only on a few computers. These computers are already used for routine clinical work, and their availability for image retrieval is limited. Moreover, most of these software packages have not been designed to extract large numbers of images and often require many clicks to retrieve images one by one.

Images come in many forms. Image quality varies among scanners, and a model trained on one scanner often does not generalize well on other scanners.

Annotations While raw images are useful for unsupervised and semi-supervised learning, the best performance is achieved by supervised learning, which requires labeled images. This raises additional difficulties.

Labeling expertise is scarce and expensive. For natural images, anyone can delineate the position of a dog, tree, or person on an image. In medical imaging, however, expertise is scarce. Most often, medical doctors are needed and their availability for these tasks is limited. In some applications, the task is easy enough for a non-medical person to be trained by a doctor and then delineate many images. Even in this

situation, annotations take time to produce and are therefore expensive.

There is no consensus over the labeling protocol. First, different clinics give different names to the same object. The bladder, for instance, can appear in various forms in the clinics: capital letters (bladder v. Bladder), language (bladder v. vessie), or additional information (bladder v. full bladder). Pre-processing the data requires naming standardization, which is time-consuming. Second, different clinics delineate structures in different ways. For the rectum, which is an OAR for prostate cancer, some radiotherapy clinics choose to contour only the wall while others contour the whole rectum. For training and validating an algorithm, however, labels must be homogeneous.

Labels are ambiguous. In natural images, there is most often consensus about the position of the object on the image. In medical images, however, there is currently no way to know the exact position of a tumor. The best prediction is a doctor's guess, but this guess is not perfect, as inter- and intra-observer studies have shown [50]. A way to mitigate this is to produce labels by consensus rather than assigning the task to a single doctor. Still, such an option is more costly and does not guarantee that the labels are correct.

2.7.2 Ethical and legal challenges

Unlike data used in other scientific areas, such as physics or chemistry, data used in the medical science are personal. This generates a tension between the need for more data to train and validate better algorithms and the obligation to maintain the privacy of personal health data. There is no correct way to solve this conflict. Different cultures with different systems of beliefs lead to different choices. In his book [100], Kai-Fu Lee argues that in China, people are more open to the idea of someone storing their personal data. The United States is still looking for a compromise. Europe has opted for the most strict option with the General Data Protection Regulation, a text limiting the gathering and exploitation of data inside the European Union.

There exists another concern about fairness between privileged and underprivileged populations. Underprivileged populations currently have less access to healthcare, and as a result the existing medical databases are biased towards the privileged populations. Access to health care is correlated to race, and people of different races do not develop the same

diseases. In particular, the incidences of different cancer types vary between whites and blacks [93]. Consequently, underprivileged populations are at risk of having less access to algorithms that suit their medical needs, thereby creating an undesirable reinforcement of inequalities.

2.7.3 Inadequate incentives

In the same way that money is the currency of the private world, citations in peer-reviewed journals are the currency of the academic world [1]. The number of times an academic researcher is cited is indeed often used as a proxy for the quality of their research. Therefore, more citations lead to more funding, which allows a head of lab to hire more PhD students and post-docs. This in turn brings even more citations, in a virtuous cycle often summarized by the maxim “publish or perish.”

While this system of incentives has generated exceptional scientific progress over the past few decades, it sometimes shows its limits when it comes to having a large group of people collaborate. As we have seen, image retrieval can be a hurdle for a radiotherapy clinic, since the staff will have to let a person use one of their limited work stations and spend a couple of weeks to extract images one by one. If the project succeeds, they have little to gain: an acknowledgment at the end of the article or a position as a middle author. If the hospital’s name is associated in the media with data loss, theft, or misuse, however, they have a lot to lose.

In this context, data are considered an asset by hospitals, either because they want to use them for their own research or because they share them with private companies in exchange for a decrease in fees for the software those companies develop. As a result, the general trend is that each hospital wants to keep its data for itself, and most works are published with small datasets. Compared with a situation where data would be more openly shared, this degrades the quality of research and ultimately slows the pace of improvement in clinical care. When papers report differences in performance, it is hard to know what part of the difference is due to the method and what part is due to the dataset. Reproducibility, a fundamental principle of scientific research, is jeopardized.

It should be mentioned that the system of incentives based on citations has not completely failed to produce large open-source databases. For example, the conference *Medical Image Computing and Computer*

Assisted Intervention Society has been launching such challenges each year for the past ten years. The release of such databases is often accompanied by a paper, which people using the data are encouraged to cite. Although those challenges go in the right direction, they are too rare and the sizes of their databases remain limited (201 annotated volumetric images were provided in the last MICCAI challenge on head and neck segmentation², for example). Other examples of large-scale scientific collaborations include the Large Hadron Collider and the OpenScience initiative from the McGill Institute of Neurology [3].

As for other commercial applications based on machine learning, there is a risk of a feedback loop forming, further complicating access to data. Today, companies often access medical data through collaboration with hospitals. As they gather more data, these companies develop better products. These better products bring them more clients, and thereby even more data. This creates the risk of concentrating the data in a few hands and decreasing competition. Less industrial and research competition is likely to slow down the pace of innovation, and therefore of improvement in clinical care. While data concentration in the hands of a few actors is not a reality for radiotherapy data today, the risk that it might further complicate the innovation of standard research lab in the future should not be overlooked.

Since the acquisition of data is financed by the community, it could be argued that the personal interest of hospitals and companies should not prevent sharing of their data. For this reason, the best way to move forward is probably to have the government take initiatives to encourage data sharing within proper legal constraints. Such an initiative has been taken in France, for example, with the creation of the Health Data Hub. A similar project is still lacking for Belgium, and while the European Union is promoting data access to the research groups benefiting from its funding [2], these efforts remain insufficient.

2.8 Domain adaptation

One way to mitigate the difficulty to obtain labels is to learn both from labeled and unlabeled examples, an approach called *semi-supervised*

²<https://www.aicrowd.com/challenges/miccai-2020-hecktor>.

learning. There exist related situations in which we wish the model to perform well on some domain (called *target* domain), while annotations for model training are mainly available in another domain (called source domain). Methods that improve the generalization on a target domain of a model trained on a source domain are called *domain adaptation* methods. Such methods can work at three different levels: at the feature-level, image-level, or label-level. We describe here each approach briefly.

Without adaptation, feature representations of source and target image tend to be different. Feature-level adaptation attempts to align these features according to some measure. One such measure is the maximum mean discrepancy [110, 161], which is defined as

$$\text{MMD} = \left\| \frac{1}{M} \sum_{i=1}^M \phi(\mathbf{x}_i^s) - \frac{1}{N} \sum_{j=1}^N \phi(\mathbf{x}_j^t) \right\|_F^2, \quad (2.25)$$

where M (respectively N) is the number of source (resp. target) domain examples, $\phi(\mathbf{x}_i^s)$ (resp. $\phi(\mathbf{x}_j^t)$) are vectors representing features associated with the i th (resp. j th) source (resp. target) domain example. Another measure of feature alignment is the difference of variances between source and target representation $\|\mathbf{C}_S - \mathbf{C}_T\|_F$, where $\mathbf{C}_S$ and $\mathbf{C}_T$ denote the covariance matrices of the source and target data, respectively [152]. Adding a term in the loss function accounting for the MMD or the difference of variances has been shown to improve cross-domain generalization. Alternatively, adversarial networks can be used [7, 53, 160]. In this setting, a first network extracts features from an input image (see Fig. 2.15). These features are used both by a label predictor, that produces a class prediction, and a domain classifier, that produces a domain prediction (i.e., “source” or “target”). The loss function is designed such that the feature extractor has incentives to produce features that lead both to (i) high accuracy for the label predictor and (ii) low accuracy for the domain predictor. In this way, similar features are encouraged for source and target domains to boost cross-domain generalization.

Image-level adaptation consists of perturbing the source images such that they look closer to the target images. This can be done with intensity-based data augmentation [125, 148] (such as brightness, contrast, or random noise), or with a cycleGAN [180]. Losses measuring

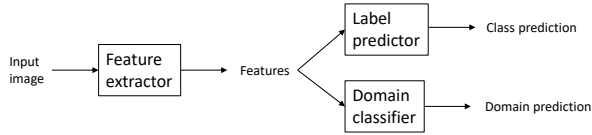


Figure 2.15: Basic framework for domain adaptation with adversarial networks.

the difference of shape after the image has been translated have also been shown to improve cross-domain adaptation [21]. PSIGAN [77] aligns the joint distribution of images and their segmentation.

Finally, domains can be aligned at the label level. In self-training, the most confident labels on target data are used for training [83, 106, 183]. Self-ensembling [51] is a method with two networks: a teacher model and a student model, where the teacher model is a moving average of the student models at different iterations. The consistency between the two models is enforced to encourage cross-domain generalization. In multi-view co-training [169], transformations are applied on the target images and one network is trained for each transformation. Consistency of the different predicted segmentations is then enforced.

Chapter 3

Secure Architectures Implementing Trusted Coalitions for Blockchain Distributed Learning (TCLearn)

In the previous chapter, we evoked the difficulty of gathering large, multi-centric databases. This is due to legal and technical challenges, as well as to a lack of incentives for hospitals to collaborate. We briefly suggested a solution: public initiatives should be taken to encourage data sharing. This chapter makes the solution more specific with a new framework based on federated Byzantine agreement. This framework has mathematical properties that are well adapted to our use case.

Let us suppose that a consortium of universities proposes to train a model with an Internet sharing platform to improve medicine. What properties should this platform have? To address the legal challenges, no contract should be needed when the consortium invites new university hospitals to join. Moreover, a member should not control which other members participate in the consortium. Also, the model should be kept confidential to create incentives for new members to provide data in exchange for access to the model. Finally, data privacy should

be guaranteed.

The proposed model based on federated Byzantine agreement has these properties. Since no member controls which other members participate in the consortium, the “federate” characteristic is particularly interesting. This specificity allows members that do not trust each other to collaborate. Indeed, the security goal 2A (see Section 3.2.2) guarantees the protection of the model against degradation by training on inadequate data. Members can collaborate without a need for a legal contract. Moreover, data privacy is ensured since the data remains in the hospitals. Finally, the blockchain distributed ledger palliates the absence of central authority.

Compared with the original paper [112], additional references are provided and concepts such as epoch and batch size are defined. The TClearn model used for the medical application (TClearn-A, which in that case is equivalent to TClearn-B) is mentioned. What is meant by “potentially sensitive” data is also specified.

Foreword

During this thesis, significant time and energy were deployed to overcome the challenges described in Section 2.7 relating to data acquisition. For each clinical partner, it involved four steps: (i) convince the head of the radiotherapy department to collaborate on the project, (ii) find agreement in the form of a legal contract governing the roles of both parties, (iii) obtain ethics committee agreement, and (iv) manually extract and homogenize the data. Seven hospitals were contacted. One hospital stopped the process at the first step, citing concerns about the potential of the project. For another hospital, the negotiations stopped at the second step after both parties could not agree on financial compensations in exchange for the data. Yet another hospital stopped at the “manual extraction” stage, mentioning privacy concerns about private data leaving the walls of the clinic. The four steps were completed with four partners. Due to the time required for data extraction, we focused on three of them. Among these, technical difficulties appeared when handling the data extracted for one hospital. Finally, data were used from only two centers: CHU-UCL-Namur and CHU-Charleroi – André Vésale. Part of the data gathered was already annotated (the

CTs), and another part was not annotated (the CBCTs). For the latter, we participated in hiring a lab technician for annotation, and we helped organize her training by a medical doctor for contouring.

I had an idea of a data-sharing framework. Its implementation required the protocol to (i) work without a legal contract, (ii) allow new members to join without the approval of other members, (iii) ensure model confidentiality, and (iv) guarantee data privacy. Luckily, colleagues were writing a paper on a similar framework. I suggested to them a concrete application to their formal method and they kindly proposed me to co-author their paper.

3.1 Introduction

Deep learning algorithms, such as the deep Convolutional Neural Network (CNN) [98], constitute outstanding predictive models that help to extract relevant information from large datasets potentially sensitive (i.e., most patients want their data to stay as private as possible). In medicine, practitioners routinely use CNN to identify various pathologies [95], contour organs at risk [102] [18], or optimize treatment plans [10]. However, the quality and size of the datasets used for the training phase have a major impact on performances. Still it is often difficult for a single organization such as a single health center to gather enough data on their own and multi-center studies are often hampered to check legal or ethical issues [41].

As a result, distributed learning [15] has been suggested for multiple applications, including in the medical field [37] [79]. This approach facilitates cooperation through coalitions in which each member retains control and responsibility over its own data, including accountability for privacy and consent of the data owners, such as patients. Batches of data are processed iteratively to feed a shared model locally. Parameters generated at each step are then sent to the other organizations to be validated as an acceptable global iteration for adjusting the model parameters. Thus, partners of the coalition will optimize a shared model by dividing the learning set into batches corresponding to blocks of data provided by the coalition members.

The naive use of a CNN in a distributed environment exposes it to a risk of corruption, whether intentional or not, during the training

phase. This is because of the lack of monitoring of the training increments and difficulty of checking the quality of the training datasets. One solution could be to have the distributed learning monitored by a centralized certification authority that would oversee the validation of each iteration of the learning process. Alternatively, a blockchain could be used to store auditable records of each transaction on an immutable decentralized ledger. This approach has been suggested in [94], where it is advocated that blockchains can be used to store signatures of patient records. In our context, blockchains would provide a more robust and equitable distributed learning process for the stakeholders involved in the learning process since all of them would also be involved in the certification process of each iteration of the model.

Weng et al. [167] proposed DeepChain as an algorithm based on blockchain for privacy-preserving deep learning training. It allows massive local training and secure aggregation of intermediate gradients among distrustful owners. Two types of user interact in DeepChain, namely, parties and workers. After signing the trading contract, parties will train the model on their own data to generate intermediate gradients. Those gradients are considered transactions and are collected according to the trading contract. Before being merged with the previous version of the weights, the collected gradients will be validated by the workers according to the processing contract. Parties are rewarded for their contributions to the training process when they provide increments, while workers are rewarded according to their contribution to the validation of these increments. Because DeepChain maintains the payoff maximization of parties and workers, a healthy, win-win environment is created. At the end of the process, Deepchain provides the auditability and confidentiality to train gradients for each participant locally while employing economic incentives to promote honest behavior. Nevertheless, it does not prevent data exposure through a malicious partner. Thus this model does not protect the shared model against degradation or divulgation.

The contribution of this paper is to derive a new model of coalitions with a high degree of reliability that respects data privacy and incentives participation in the coalition without a central authority. In order to implement this model, we propose novel scalable security architectures, called Trusted Coalitions for distributed Learning (TCLearn), based on either public - to be open to a large amount of participants - or permis-

sioned blockchains to provide distributed deep learning with increasing levels of security and privacy preservation.

In our approach, a CNN model is shared among the members of the coalition and optimized in an iterative sequence, with each member of the coalition updating it sequentially with new batches of local data. Each iteration of the shared model is validated by a process involving the members of the coalition and then stored in the blockchain. Each step of the evolution of the model can be retrieved from the immutable ledger provided by the certified blockchain.

While early implementations of blockchains, such as Bitcoin, initially relied on a proof-of-work [45] consensus mechanism, other architectures such as proof-of-stake [84] have been suggested since. Our approach has a different goal: to provide an iterative certification process for each learning step of the shared model, all of them being registered in the underlying ledger. We therefore suggest a new consensus mechanism based on a custom Federated Byzantine Agreement (FBA) [96] integrating performance evaluation into the block validation step.

In this article, the model designates a (deep-)CNN with its associated weights (parameters). The gradients represent the evolution of the model weights after a training step. The supervisor is an entity handling the storage of the model and controlling its access.

3.2 Threats and Security goals

In this section, we discuss threat scenarios where the data or model is attacked by disruptive parties. In comparison with DeepChain, we do not have trust issues with the encryption process because it is done by a trusted entity: the supervisor. This role is strictly restricted to ensure secure channels between the members of the coalition. For this reason we focus on threats applied to the data and the model.

3.2.1 Threat 1: Keep control over the data

The leakage of data is an important threat to address, specially in the case of medical data. New regulations (e.g., the GDPR in EU) are making access to personal data more restrictive in order to respect their owners' privacy and consent of use.

Security Goal 1: Privacy of the training dataset

This threat refers to two different elements: a) disclosure of the private data brought by each partner to perform a training step and b) the possibility of reconstructing part of the training set from the generated gradients, which is known as the long-term memory effect [151].

The use of distributed learning protects the training data. It makes it possible to improve a shared model with new data without their leaving the local environment.

To avoid any leaks from the gradients, differential privacy has been proposed in the literature. A computation is considered to be differentially private if the probability of producing a given output does not depend very much on whether a particular data point is included in the input dataset [46]. For any of two datasets D and D' differing by a single item, also called adjacent databases, and any output O of function f ,

$$Pr\{f(D) \in O\} \leq e^\epsilon Pr\{f(D') \in O\} \quad (3.1)$$

The parameter ϵ controls the trade-off between the accuracy of the differential privacy f and how much information it leaks.

In our use case, as evoked by Shokri et al. [147], participants may reveal some information about the training datasets indirectly via public updates to a fraction of the CNN parameters during training.

Several approaches have been proposed in the literature to mitigate this effect. As an example, the use of Homomorphic Encryption (HE) techniques has been considered in [8]. In their article Shokri et al. [147] suggest sharing a small proportion of their gradients randomly and perturbing them by adding some noise. This approach reduces the performances of the model while improving privacy.

3.2.2 Threat 2: Keep control over the model

This threat refers to two different elements: a) the model is exposed to a risk of corruption or degradation during the training phase. For instance, a partner could attempt to train the model on corrupted data or a different pathology from the one studied. b) The blockchain keeps track of each and every modification to the model, but does not prevent unauthorized use of the shared model outside the coalition. If the

learned model has to be confidential, an extra level of protection must be added to prevent any potential deliberate or accidental leak outside the consortium.

Security Goal 2a: protection of the model against degradation by training on inadequate data

The evolution of the model must be resilient to malicious or clumsy actions that would decrease the performances of the model. The proposed approach must detect this kind of misuse and reject the resulting increment. Traceability of every operation involving the model must be ensured to deter any malicious event.

Security Goal 2b: confidentiality and traceability of the model

The disclosure of a trained model defined as confidential by the consortium leads to a leak of intellectual property. This threat depends on the confidentiality level required by the consortium.

3.3 A scalable security architecture for trusted coalitions

The trade-off between security and cost [47] is a well-known issue. In this section, we will develop three different methods corresponding to three distinct trust levels depending on the shared rules in the coalition:

- **Method TClearn-A:** The learned model is public but each member of the coalition is accountable for the privacy protection of its own data.
- **Method TClearn-B:** The learned model is private (shared only within the coalition) and the members of the coalition trust each other.
- **Method TClearn-C:** The members of the coalition do not trust each other and want to prevent any unfair behavior by any of them, such as unauthorized use or leaking of the model outside the coalition.

These three methods address the previously described security issues at different levels (see Table 3.1), offering an inherent trade-off between security needs and costs.

Table 3.1: Summary of the features for the three TClearn methods.

TClearn	A	B	C
Data privacy	✓	✓	✓
Protection against degradation	✓	✓	✓
Model privacy for outside threats	✗	✓	✓
Model privacy for inside threats	✗	✗	✓
Model access audit trail	✗	✓	✓
Model leakage traceability	✗	✗	✓

3.3.1 Architecture of TClearn-A

Type of coalition: coalition sharing a public model built using private datasets. The integrity of the increments is ensured through the use of a new federated Byzantine agreement protocol.

Solution to Threat 1 (Privacy of the training dataset)

A partner willing to improve a model with its new data has to:

1. fetch the current version of the weights W_i and apply it to the known architecture.
2. trains the model locally with its own dataset, generating new gradients G_{i+1} .
3. uploads the resulting gradients G_{i+1} .

This way, the dataset used for training never leaves the partner's infrastructure, ensuring its privacy and excluding any leakage of the processed data. Because access to the previous datasets is forbidden, this training step builds epochs only from the new dataset provided by the user. After this training, the generated gradients G_{i+1} are uploaded and applied to the previous weights (W_i), leading to new parameters (W_{i+1}) and a candidate model, noted $tmp(M_{i+1})$.

To protect against any leaks coming from the gradients, it is possible to share a small proportion of the gradients and add some noise as

suggested by Shokri et al. [147] in order to improve differential privacy. The parameter ϵ , controlling the trade-off between the accuracy and privacy, should be determined by the user.

Solution to Threat 2a (Protection of the model against degradation by training on inadequate data)

Our approach relies on a blockchain to carry only unalterable cryptographic hashes of the successive training steps of a model built in a distributed environment that ensures the validation of the successive iterations. The iteratively optimized model is made public. Each block represents an iteration step achieved locally by a specific member of the coalition and validated by the whole coalition. First, the model and the genesis block are initialized, setting the architecture (layers, activation functions, loss function, etc.) and the weights according to a normal distribution. The weights of the model are updated iteratively by the batches of data provided by the members of the coalition.

In our approach, we use a blockchain relying on a federated Byzantine agreement to prevent corrupted increments caused by inadequate training from being added to the model. The candidate increment has to be validated by multiple validators. Since the blockchain and the deep learning model are strongly linked in TClearn, the FBA has two goals:

- control the quality of the updated CNN model, through a “peer review” system and
- control the integrity of the new block (hash, index, timestamp, etc.) and concatenate it to the chain.

The FBA process starts with the random selection of validators within the consortium. All the members can be selected but the probability depends on the size of the consortium and the proportion of data brought by each member as follows (Equation 3.2),

$$S_i = 1/N + D_i/D \quad (3.2)$$

where S_i is the strength of the partner i , N the total number of partners in the consortium, D the total amount of samples used for the model and D_i the amount of samples supplied by the partner i . Initially, this strength is the same for each of the partners and evolves with each

contribution. The main role of the validators is to check the candidate model M_{i+1} incremented from M_i proposed by a partner. The model M_{i+1} must show an improvement over the previous one. The test on the validity of the increment should protect the model against corrupted data in the training set.

Two types of test databases are used to assess the performances of each increment of the model (Fig. 1) and to avoid the introduction of invalid or inadequate training sets:

- A global test database (G), common for every block creation and for all the partners. This database is created by experts to be representative of the pathology.
- A local test database (L), different for each partner in the consortium. To avoid overfitting on the global test dataset (G), a small percentage of the input signals is put aside locally for each contribution. It is used later by each validator as a local test set to evaluate the proposed model individually.

Both datasets are used for the testing phase. First, results obtained using the common, global dataset are compared between validators to ensure that the candidate model is functional and identical. Then, those "global" results are merged with those obtained using the individual, local datasets. To be accepted, the candidate model must have higher performances than the previous model within a specific threshold λ ($\in [0, 1]$) (Equation 3.3).

$$\text{Block creation IF } \lambda \times perf(M_i) \leq perf(M_{i+1}) \quad (3.3)$$

Once the model is accepted, a new block can be created. First, the block creation requires that a "general" (or "speaker") be selected from among the validators. The "general" will be the creator of a new block containing the reference to the validated model M_{i+1} and the ID of the partner who proposed it. All the validators then check the integrity of this candidate block.

This block is analyzed in the frame of a delegated Byzantine fault tolerance system also suggested by Damaskinos et al. in [36]. Each validator broadcasts its opinion (acceptance or rejection) of this block to the other validators. If at least two-thirds of the validators agree,

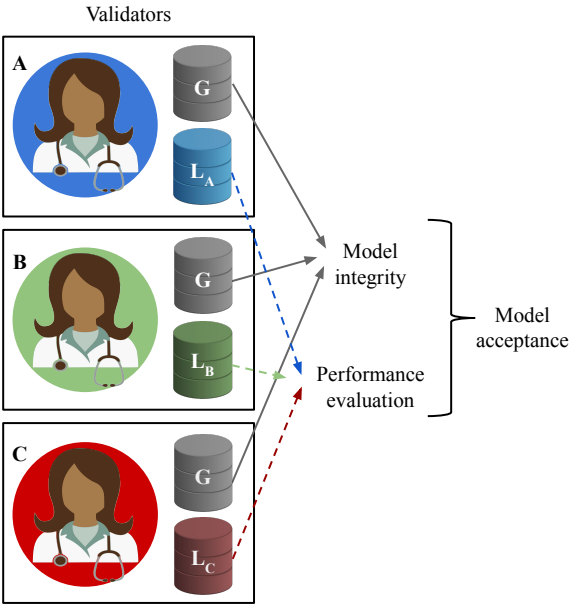


Figure 3.1: Federated Byzantine agreement and candidate model checking process. Two datasets are used: a global one (G) similar for all the partners that is used to control the model’s integrity and a local one (L), different for all the partners that is used for performance evaluation. After a majority vote, the candidate model is accepted or not.

the FBA process can stop, leading to the acceptance or rejection of the block. If not, the role of "general" is switched to another randomly selected validator and the block creation process can restart. If the block is accepted by the validators, the "general" can append it to the blockchain and broadcasts this update to the whole consortium, requesting synchronization of the blockchains.

The overall scheme of TClearn-A is represented in Fig. 2.

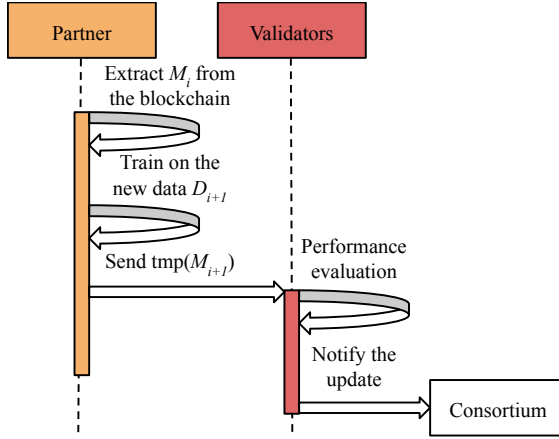


Figure 3.2: Scheme of the TClearn-A procedure. Partner trains a model (M_i) on its data (D_{i+1}). The candidate model $tmp(M_{i+1})$ is validated by a federated Byzantine agreement.

Each block includes:

- block index
- timestamp
- previous block hash
- hash of the model's parameters
- user ID: identification of the contributor
- users' strength: level of contribution of each member to the FBA
- block hash

Input: A blockchain bc and a training set D_i

```

1 modelTrain( $bc, D_i$ ):
2    $w_i = \text{loadWeights}(bc)$ ;
3    $M_i = \text{initializeModel}(w_i)$ ;
4    $tmp(M_{i+1}) = \text{training}(D_i, M_i)$ ;
5    $tmp(bloc) = \text{blocCreation}(bc, tmp(M_{i+1}), D_i)$ ;
6 federatedByzantineAgreement( $bc, tmp(bloc)$ ):
7    $\text{controlPerformances}(bc, tmp(M_{i+1}))$ ;
8 if  $\lambda \times \text{perf}(M_i) \leq \text{perf}(M_{i+1})$  then
9   | if  $\text{integrity of } bc == \text{TRUE}$  then
10  | |  $\text{blockCreation}(bc, tmp(bloc))$ ;
11  | end
12 end

```

Algorithm 1: TCLearn concept.

The pseudo code for this approach is given in 1.

Audit & traceability of the training data

The evaluation of performances may not be enough to avoid duplicate input signals during training. For this reason, anonymous IDs can be attached to each input signal and stored in the blockchain. This process also enforces the auditability and the traceability of the input signals. This can be implemented as an additional field stored in each block, such as an anonymized data ID, therefore avoiding training on duplicate input signals.

3.3.2 Architecture of TCLearn-B

Type of coalition: coalition sharing a private model built using private datasets in a restricted consortium of trusted partners. The integrity of the increments is ensured through the use of the FBA protocol. In this situation, the model has to be protected during transfers between partners and for its storage.

Solution to Threat 1 & 2a

As this approach is based on TCLearn-A, the data privacy of the inputs is preserved and the model iterations are validated by members of the coalition.

Solution to Threat 2b (confidentiality and traceability of the

model)

With TCMlearn-A, the evolutions of the model are certified by the blockchain. However, this scheme does not guarantee the confidentiality of the model during its distribution to partners. This situation is not acceptable in some situations where the privacy of the model must be preserved (for example, if the members of the coalition want to avoid any leakage of the model).

To solve this issue, the storage, transfer, and upgrades through gradient computations of the model have to be protected by encryption, where the private keys are stored by some trusted entities. In this work, we propose to isolate all iterations of the model in an external, off-chain encrypted storage facility ("vault") and control its access by each partner.

We also suggest the use of secure transport (e.g., TLS or S/MIME) for transferring the model. Moreover, the model could be stored by using an efficient encryption method and implementing access control and auditing mechanisms. Only authorized users should be able to download a given version of the model weights, and each access should be logged in an audit trail.

In order to (1) ensure that only authorized participants are granted access to the models and (2) minimize the risks of leakage, we propose to store the actual models (including the associated weights) in an external and secure "off-chain storage vault". In this approach, the blockchain provides only "links" to the corresponding version of the model's weights.

This introduces a single point of failure (the secure, off-chain storage and its associated access control infrastructure), but offers three major advantages. First, it greatly reduces the size of the blockchain while increasing its scalability, so that each participant needs to synchronize fewer data. Second, this approach makes it possible to implement active access control over all of the stored information (e.g., who is allowed to access which kind of data, when and how). Finally, each of the requests to access the secure, off-chain storage vault could be recorded in a journal, offering the ability to audit all accesses to any record. This audit could be used to gather statistics about the actual accesses of each individual user to models. This in turn could be used to restrain future accesses (e.g., allowing users to access a certain number of models

depending on their level of contribution to the models). Therefore, a moderated level of traceability is offered: if a given version of a model is leaked, it will be possible to audit all of the requests related to this specific version predating the leak in order to establish the list of partners that actually downloaded this version and potentially leaked it.

The blockchain/offline storage approach requires that an entity (the "supervisor") manages the access control and the secure storage of the model. An overview of this approach is presented in Fig. 3.

If we look again at the pseudocode (cf. Algorithm 1), we can add a line of encryption between line 5 and 9, and one of decryption between lines 6 and 7.

Secure authentication and transport of the model

The secure authentication and transport of the model could be performed in two ways: either online (requiring direct, interactive communication between the partner and the supervisor) or offline (allowing for delayed communication, e.g., using periodic batches of file transfers or e-mail).

In the former case, we strongly recommend using an industry-standard protocol such as TLS (Transport Layer Security) v.1.3 (RFC 8446), which is used, for instance, in HTTPS. Unlike network-level encryption (such as IPsec VPNs), TLS offers end-to-end encryption that guarantees the confidentiality of the model and gradients, including between the secure gateway (e.g., VPN concentrator) and the machine performing the machine learning. The server (the supervisor) must be authenticated by validating its X.509 certification chain. The identity of the client could also optionally be requested (using a X.509 certificate chain) in order to provide a stronger authentication mechanism. Once the TLS handshake and authentication successfully performed, the model and its gradients can be exchanged safely using strong encryption (such as AES-256).

In the latter case, we suggest using an industry-standard protocol such as S/MIME (RFC 5751). Both the supervisor and the partner need to possess X.509 certificates, which could be used to both authenticate (digital signatures of the partner's request and the supervisor's reply) and encrypt the data (using the partner's public key so that only this partner can decrypt it). In this scenario, the partner could send a request, signing it digitally using its X.509 private key; the supervisor ensures the authenticity of the request by verifying the provided digital

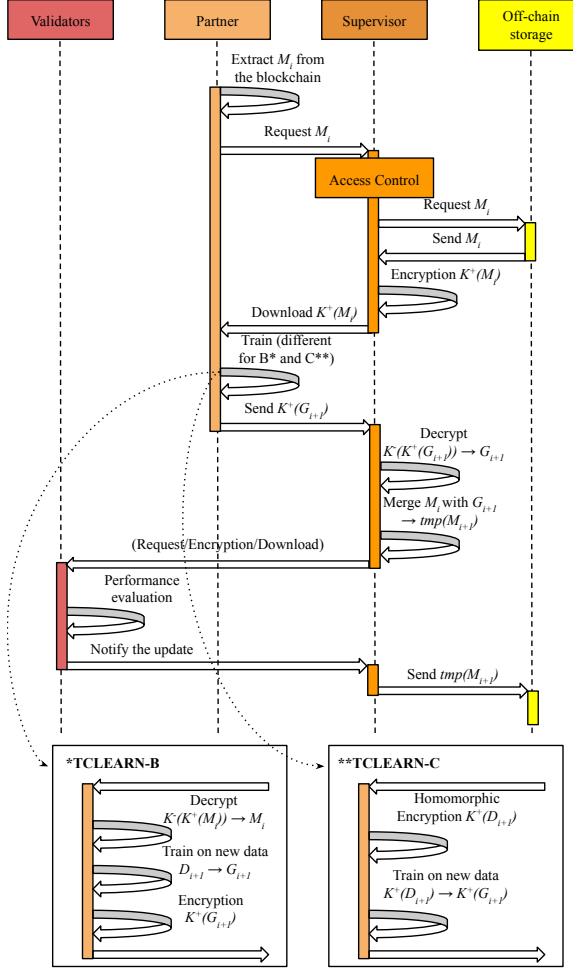


Figure 3.3: Scheme of the TClearn-B & C procedure. The partner trains a model (M_i) on their data (D_{i+1}) leading to new gradients (G_{i+1}). The model is successively encrypted and decrypted, e.g., using a public (K^+) and a private key (K^-), respectively. The candidate model ($tmp(M_{i+1})$) is validated by federated Byzantine agreement. The two methods are principally differentiated by the training process in or out of the encryption domain.

signature (decrypts it using the partner’s public key). Once the partner (and the corresponding request) are authenticated, the supervisor will be able to send the encrypted model and gradients. This operation could be performed by the supervisor, for example, by first generating a random session key that could be used as a symmetric key with a symmetric encryption system such as AES. This random session key is then encrypted using the partner’s public key (so that only the partner could decrypt it using its private key) and sent along with the encrypted model and gradients to the partner. The partner then simply needs to decrypt the symmetric session key using its private key and then use this (decrypted) session key to decrypt the message. This scheme (illustrated in Fig. 3) requires each party to possess an X.509 certificate and the associated private key in order to decrypt information provided by other parties.

3.3.3 Architecture of TCLearn-C

Type of coalition: coalition sharing a private model built using private datasets in a restricted consortium of untrusting partners. The integrity of the increments is ensured through the use of an FBA protocol. To protect the model in this situation, it is necessary to secure the exchanges and the storage even at the partners’ facilities.

Solution to Threat 1 & 2a

As this approach is based on TCLearn-A, the data privacy of the inputs is preserved and the model iterations are validated by members of the coalition.

Solution to Threat 2b (confidentiality and traceability of the model)

In this scenario, our objective is to identify the partner responsible for leaking a model. Such identification could be performed by adding some unique, hidden information to the model provided to each partner, e.g., by altering the weights by introducing a moderate noise following a specific, hidden pattern (constituting a watermark) every time the model is requested by a partner. The date of the access, identity of the partner, and associated hidden pattern (the watermark) could then all be stored in an audit trail allowing to uniquely identify the partner associated with a leaked model. Unfortunately, CNNs are quite robust to a slight alterations of the weights, which means that an adverse party might try to

alter those (watermarked) weights, jeopardizing the recovery of the watermark while keeping the model usable. This adverse party could even perform a subsequent training of the model on new datasets, further compromising the watermark’s recoverability.

Another option could be to send to each partner a model encrypted with a specific key, allowing them to manipulate (use and even train) it without being able to decrypt it. Some crypto algorithms (such as the ElGamal scheme [48]) allow operations to be performed directly on the encrypted data without knowing the associated private key. For instance, the ElGamal scheme offers an encrypted product operator allowing one to compute the product $E(m_1 \cdot m_2) = HM(E(m_1), E(m_2))$ of two encrypted messages $E(m_1)$ and $E(m_2)$. An operator offering such a property constitutes a homomorphic operator. An encryption algorithm offering both homomorphic product and homomorphic addition, such as the Brakerski-Gentry-Vaikuntanathan scheme [17], constitutes a fully homomorphic encryption scheme.

Homomorphic encryption brings a new level of protection to the model (since actually only the supervisor is able to decrypt it), but also a considerable drawback: in order to perform prediction, the encrypted result must be sent by the partner to the supervisor to be decrypted and the final result sent back to the partner. This reduces the partners’ autonomy (they depend on the supervisor to perform any prediction using the model) and potentially introduces a single point of failure.

Similarly to TClearn-B, the confidentiality and the distribution control over the models is ensured by off-chain storage of the models and gradients. However, since the members of the consortium do not trust each other, no member should have access to unencrypted models or gradients. This is ensured by the use of the aforementioned homomorphic encryption technique. In order to use the model (prediction), the input signals must first be encrypted using the same technique and the same homomorphic public key (which must be sent together with the encrypted models and gradients to each partner). This public key can be used only to encrypt (and not decrypt) data. Once the prediction operation (using encrypted model and input signals) is performed, the result must be decrypted. This must be done by the supervisor (which holds both the homomorphic public and private keys corresponding to the models sent to each partner). The supervisor uses its homomorphic

private key to decrypt the result, which is then sent to the requesting partner. The supervisor must inspect the "results" to decrypt very carefully in order to reject any attempt to decrypt models or gradients.

Considerations regarding the use of homomorphism in CNNs

In TClearn-C we suggest the use of homomorphic encryption to ensure traceability and confidentiality of the model. In this scenario, the whole manipulation of the CNN takes place in the homomorphic domain, which introduces some challenges that are described hereafter.

First, the neural network algorithm requires that both addition and multiplication operations be used and combined. Consequently, the "somewhat homomorphic encryption" scheme (see ElGamal [48], BGV [17] or Paillier [132]), which allows the use of one sole type of homomorphic arithmetic operation, cannot be used. We thus have to use FHE, which requires considerable memory and computational resources.

The use of a CNN in the homomorphic domain also introduces implementation issues. Indeed, while most FHE implementations support only homomorphic addition and multiplication, implementing CNN activation functions in the homomorphic domain requires complex operations such as trigonometric operations ($\tan H$), exponentials (sigmoid), and tests (ReLU). Zhang et al. [176] proposed the use of a Taylor development to replace the sigmoid function in order to compensate for the lack of exponential functions in FHE schemes. Although this method allows us to use the sigmoid function in the homomorphic domain, it remains approximate. Moreover, it still increases the number of operations to perform.

Solving those problems has been the subject of some recent work. As an example, in [13], Bourse et al. presented a framework for homomorphic evaluation of neural networks using a highly optimized FHE algorithm. This scheme, dubbed TFHE, offers several orders of magnitude performance improvements over previous FHE architectures. In this article, the authors used a "discretized" neural network to allow the creation of a model capable of fitting data from the MNIST database. Several tips are proposed to reduce the time required for learning and prediction (bootstrapping, look-up table, and noise management).

One could thus use Bourse's solution. If this approach is not sufficient to mitigate the resource issues associated with TClearn-C, the training process could alternatively be performed using tamper-proof

black boxes deployed in each of the partner’s facilities. Such black boxes could be used to decrypt the CNN model, retrain it, and then re-encrypt it without anyone being able to interfere. However, this solution requires all of the partners to request the installation and maintenance of this black box by a trusted third-party service provider.

If we look again at the pseudocode (cf. Algorithm 1), we can add a line of encryption between lines 5 and 6.

3.3.4 Additional features

In addition to the TCLearn approaches, we propose two optional extra features.

Reverting to a previous state of the model

Each and every increment of the model recorded on the blockchain and validated using the FBA are accessible by the partners. In the case of late detection of corruption, it could be necessary to restore the model to one of its previous states. For this reason, we add the possibility of performing prediction or even continuing the training from old weights stored in the off-chain site. Only authorized partners are able to perform this operation, in which case a new block is generated, prevailing over previous learning steps.

Updates to the model hyperparameters

The model architecture (kernel size, number of layers, etc.) is initialized in the genesis block of the blockchain but some hyperparameters (epoch number, batch size) can be modified during the learning step. In deep learning, the *epoch number* is the number of complete pass on the whole training data performed by stochastic gradient descent. The number of samples used per iteration (in other words, per parameters update) is the *batch size*. The learning rate of the optimizer for the gradient descent is a critical parameter. It could be high at the beginning of the blockchain because the model is still naive, but the more precise the model is, the lower the learning rate should be (without becoming too low). A method that adapts the learning rate according to the performances of the model automatically could be integrated into our concept [43].

3.4 Security analysis

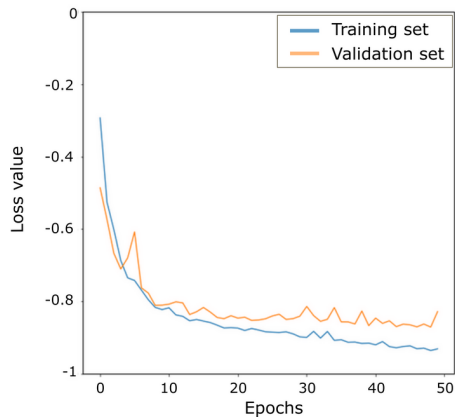
3.4.1 Solution to Threat 1: Keep control over the data

Distributed learning for a medical application: an example

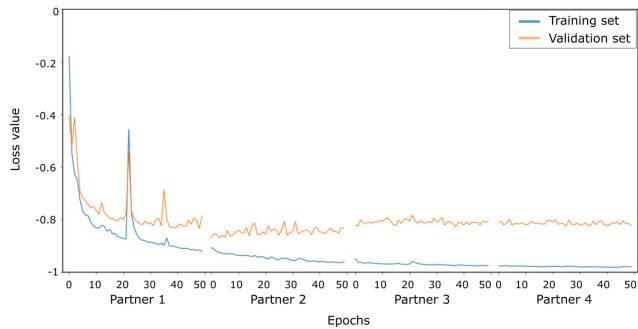
To illustrate the use of distributed learning, we propose to apply it to a medical challenge that has already been solved using CNNs, namely, bladder contouring on computed tomography scans [102] [18]. Léger et al. proposed using U-Net [143] to segment the bladders of 339 patients with prostate cancer. Their semi-automatic approach takes two channels as input (a 3D volume of the bladder with one of its slices labeled manually by an expert) and outputs a prediction for the target bladder segmentation tile.

We reproduced their results using the parameters and database used by Léger et al. [102] [18]. First, we performed a centralized training using all the training samples in 50 epochs (see Fig 3.4.1a). Then, we randomly split the initial database into several subsets to simulate several partners and perform a distributed learning using the smaller datasets successively. This follows the framework of TClearn-A (which for this use-case is equivalent to TClearn-B, i.e. there is no homomorphic encryption). Thus, each partner realizes their own training process (with 50 epochs) one after the other (see Fig 3.4.1b).

Once the training was over, the accuracy achieved with a distributed database was not significantly different from that resulting from centralized training (88.4% and 88.7%, respectively). This result means that the model is able to catch relevant information from the split dataset as well as from the centralized one. This result is the same even if we switch the order of the partners. In both cases each partner has a balanced dataset. Otherwise, this could lead to performance impairment of the model; TCLEARN would therefore reject the proposed model. Moreover, to avoid the long term memory effect, the training step included batch inputs. Thus, each gradient represents the average information provided by these batches and not just by a single patient. Therefore, we cannot directly infer the features of a given patient from the gradients.



(a) Computation of the first neuron of a feature map.



(b) Computation of the second neuron of a feature map.

Figure 3.4: Loss value (- Dice score) during training (a) in a centralized and (b) in a distributed way [102] [18].

3.4.2 Solution to Threat 2: Keep control over the model

Solution to Threat 2a: protection of the model against degradation by training on inadequate data

To protect the model against malicious events that might reduce its accuracy we proposed using of federated Byzantine agreement. This system ensures the integrity and performances achieved by each proposed candidate model.

Solution to Threat 2b: confidentiality and traceability of the model

This threat depends on the confidentiality level required by the consortium. In TClearn-A, the model is open and does not require any protection. In TClearn-B, the model is built using private datasets and shared within a restricted consortium of trusted partners. In this case, we propose using secure transport for exchanging the model between the partners and the supervisor. For TClearn-C, the model is built using private datasets and shared within a restricted consortium of untrusting partners. In this case, we suggest the use of homomorphic encryption allowing to use the model in the encryption domain.

In both cases, the encryption is managed by a trusted supervisor (either at the transport level such as in TClearn-B or at the application level such as in TClearn-C). Moreover, we suggest storing the model in an external off-chain storage vault. The blockchain will contain links to the model, thereby minimizing its size while allowing the implementation of access control and auditing mechanisms. Only authorized users should be able to download a given version of the model weights, and each access should be logged into an immutable audit trail.

It might be possible to use this audit to gather statistics about the actual accesses of each individual user to models. This in turn could be used to restrain future accesses (e.g., allowing users to access a certain number of models depending on their level of contribution to the models). Therefore, a moderated level of traceability is offered even in the case of TClearn-B: if a given version of a model is leaked, it is possible to audit all of the requests related to this specific version predating the leak in order to establish the list of partners that actually downloaded this version and may have leaked it. In the case of TClearn-C, the public

homomorphic key associated with the model could be directly matched to a single access (member, date and time, and accessed model).

3.5 Implementation and evaluation

In this section, we present an implementation prototype for TCLearn-A & B, that demonstrates the feasibility of our approach. This prototype is available at <https://github.com/slugan/TCLearn>.

The goal of this prototype (which is not intended for deployment in a production environment) is to illustrate and demonstrate an example of implementation of the presented architectures.

The blockchain architecture is based on the proof-of-concept implementation proposed by Gerald Nash in his article¹.

Our prototype relies on Python version 3.6.8 and Numpy version 1.16.4. The server-side scripts (including the supervisor) are implemented using Flask Microframework version 1.1.1 while the client-side scripts rely on requests version 2.22.0 (a Python HTTP library). The sample certificates are generated using OpenSSL version 1.0.2k. The (purely illustrative) server-side TLS encryption is handled by the uWSGI application container server version 2.0.17.1. The deep learning part is based on Tensorflow version 1.5.0 and Keras version 2.2.4.

In this prototype, each virtual site is associated with a Flask application. A separate application simulates the supervisor (in the case of TCLearn-B and TCLearn-C). Python scripts are used to send commands to these applications using HTTPS calls (e.g., to simulate submissions by a given partner). Please consult the documentation on the GitHub repository for more information.

This proof-of-concept illustrates the principles of the TCLearn architectures on the MNIST dataset (32×32 pixels handwritten digits, 60000 training images and 10000 test images). The test set is split in two groups: the first half is used as a global test set and the second half, divided equally between 7 partners, constitutes the virtual consortium. Each partner is able to test a candidate model using both the global test set and their own local test set. The training set is split into batches of increasing size, each associated with a distinct partner

¹<https://medium.com/crypto-currently/lets-build-the-tiniest-blockchain-e70965a248b>.

of the consortium and constituting the input used by this partner to submit a new candidate model (iteration). The CNN architecture used is: *Input* $\rightarrow$ *Conv* $\rightarrow$ *Maxpool* $\rightarrow$ *Fully Connected* $\rightarrow$ *Output*. The optimizer is Adadelta with a learning rate of 1.0, a mini batch size of 64 and number of epochs equal to 20.

In the first example (Fig. 3.5), the training set is randomly and arbitrarily split in (small) batches of 204, 1080, 1080, 3000, 4800, 4800 and 9600 images (40% of the complete set). The accuracy slowly increases (due to the small size of the initial batches) quite steadily among the various test sets with each iteration (corresponding to the submission of a new candidate model by a partner). The global test database (7 times larger than each of the local test set of each partner) is associated to the highest accuracy values as expected.

In the second example (Fig. 3.6), the training set is now randomly split in batches of 2400, 4800, 4800, 6000, 10800, 10800 and 20400 images (100% of the complete set). Given the larger size of the initial batches, the accuracy reaches a high level from the firsts iterations, and globally continues to increase. However, we notice a few occurrences of decreasing accuracy from one iteration to the next one when tested against some local test sets. This illustrates the need for the joint use of the global test set by each partner in addition to their own local test set when evaluating a candidate model. This also shows the importance of the λ threshold tolerating a moderate decrease of the performances when evaluating a new submitted model.

3.6 Conclusions

In this article, we propose a new architecture for distributed learning by a model based on federated Byzantine agreement mechanism. The performance of the model is ensured through a shared evaluation of individual contributions leading to acceptance or rejection based on an objective criterion.

This approach makes it possible to constitute trusted coalitions in which the actions for updating the model by the members are registered in a public ledger implemented as a blockchain. We have explored three kinds of coalitions according to the access control required for distributing the model. Each approach corresponds to distinct trust levels

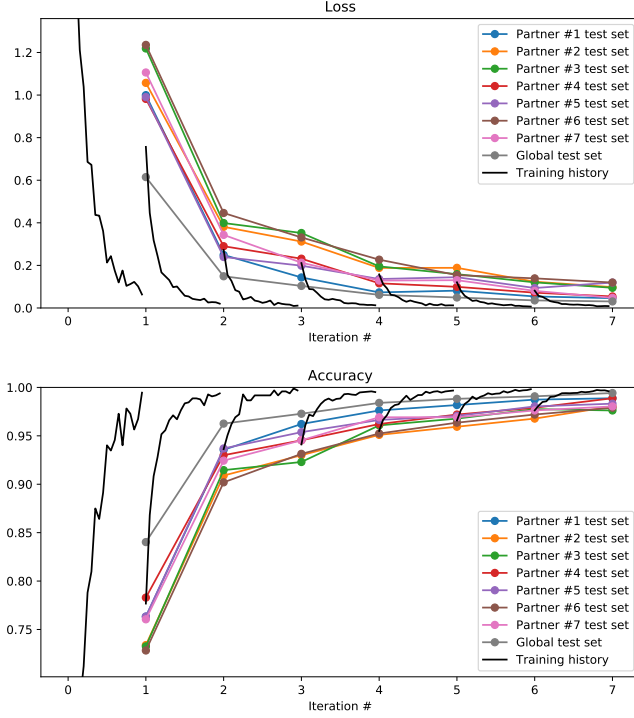


Figure 3.5: Loss and accuracy with small batches (204, 1080, 1080, 3000, 4800, 4800 and 9600 images). Training history represents the performances obtained during the training process on the training dataset.

that depend on the shared rules in the coalition. We have proposed solutions that rely on efficient cryptographic tools, including homomorphic encryption. We have given an example of our proposed architectures with the case of the distributed learning by a CNN model applied to distributed medical images databases. The proposed architectures safeguards data privacy thanks to a system of encryption and off-chain storage to avoid the dissemination of sensitive medical data or metadata.

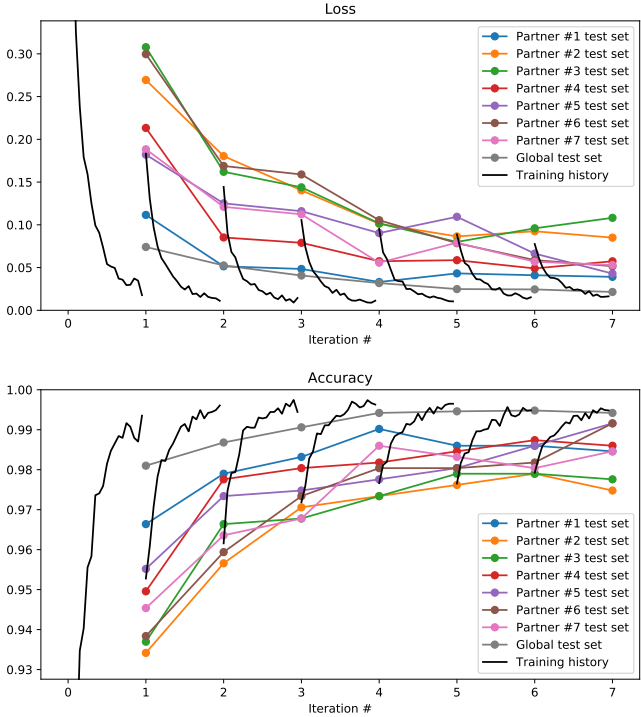


Figure 3.6: Loss and accuracy with larger batches (2400, 4800, 4800, 6000, 10800, 10800 and 20400 images). Training history represents the performances obtained during the training process on the training dataset.

Chapter 4

Contour Propagation in CT Scans with Convolutional Neural Networks

In Chapter 1, we saw that treatment planning was still a slow manual process involving the manual contouring of structures in volumetric images. In Chapter 2, we showed how deep learning could be used to train models for automated segmentation. In this chapter, we investigate a new method for contouring organs at risk in planning CT faster, while keeping acceptable accuracy thanks to *knowledge propagation*.

We specify here an issue about slice sampling that is not discussed in the original article. Typically, CT scans have a slice every 2 to 3 mm. So, over ten new (1 mm) slices there are about three original contours and seven linearly interpolated contours. The argument could be made that our network is just learning to do a linear interpolation. However, what is done here is different since the network takes as input information about one prior slice only, while an interpolation requires two contours. Indeed, the target slice is often a slice that has been produced based on the prior. We could therefore imagine that prior and target slices are more correlated than what would be observed if we were working in the original resolution of each machine. This could have favored our algorithm compared with the situation in which it is aimed to be deployed. Even though it is an interesting point to keep in mind while reading this chapter, studying the importance of this issue is beyond its

scope.

In this work, eleven networks were trained from scratch. There is redundant information in the weights of the different networks. Therefore, transfer learning might have accelerated the training. This possibility has not been investigated in this study and could be part of future work.

For the CNN without prior, Table 4.1 reports the same DSC and JI for all inter-slices distances. What is meant is that the CNN without prior information takes as input the image only, and thus the inter-slices distance is not defined in this case. The results that are present in the different columns come from one unique experiment.

An extension of this work could consist of building deep learning-based patient-specific models. In this configuration, the contours of a given patient could be added to the training set during the treatment, and a new model could be learned. This would, however, raise two difficulties: (i) manual contours must be provided for a few CTs of this patient, and (ii) a new model must be learned specifically for each patient (which is time-consuming). Deep learning-based patient-specific models are left for future work.

While the results presented in this chapter use only the contours of the prior slice, the image of this prior slice could have interesting predictive value as well. In experiments not described in the chapter, we tried to add the prior image as an additional input to u-net, but we did not observe any performance improvement. Investigating why this is the case is left for future work.

Compared with the original article [102], references have been added for the libraries that have been used. Several choices in terms of learning and inference (i.e., architecture, probability map threshold, data augmentation) are now motivated.

4.1 Introduction

Computed tomography (CT) image segmentation is highly useful for a large number of medical applications such as image analysis in radiology [117], treatment planning and patient follow-up in radiotherapy [146] or therapeutic response prediction through radiomics [22]. Classical clinically used segmentation methods include atlas-based methods [73] or active contours [34], but are generally failing in cases where the regions of

interest are subjected to large deformations or matter income/outcome. Statistical shape models [67] allow to capture shape variations but require landmarks definition. Pixel-wise classification has also been performed with random forest and SVM classifiers [136], [113], potentially completed by structure enhancing methods such as conditional random fields [70] or graph cuts. However, those methods require a tedious and subjective definition of features. Alternatively, numerous hybrid methods have proposed to combine several complementary segmentation approaches [173], [54], [157], [128]. Nevertheless, none of them closes the challenging task of CT image segmentation.

In parallel, the recent advances in computing capabilities and the availability of representative datasets allowed deep learning approaches to reach impressive segmentation performance, competing with state-of-the-art segmentation tools in many fields of medical imaging [107]. The main advantage of deep learning with respect to the aforementioned approaches is its high versatility and its capability to learn automatically a complex input-output mapping through the tuning of a large number of inner parameters. Fully convolutional neural networks [109] gained in popularity compared to patch-based approaches since they benefit from a higher computational efficiency and a better integration of contextual information. In particular, the u-net fully convolutional neural network architecture [143] resulted in good segmentation performances on 2D medical images, as well as its 3D extension [31] and 3D V-Net variation [120].

Deep learning segmentation algorithms have been successfully used to segment structures on CT images in the head and neck [72], the pelvic [82] and the abdominal [144], [97] regions. Those deep models rely on the assumption that the learned filters are able to identify automatically, from the training data, the features that are relevant for the task at hand. However, in many applications, additional information, also named *prior information*, is available and should be exploited to constraint the segmentation process towards the desired output. Recently, shape prior constraints have been incorporated in deep neural networks through PCA [121], resulting in an improved segmentation robustness and accuracy. Anatomical constraints have also been accounted for through the combination of several deep neural networks [159].

Similarly, the availability of a coarse or approximate segmentation

may provide a valuable prior information to guide a fine-grained segmentation process. Examples of practical applicative scenarios in which such approximate segmentation is available include dense 3D segmentation (for which only a subset of the slices have been segmented), but also the adaptation of radiotherapy treatments (for which the segmentation of an initial image should be propagated to subsequent images acquired along the treatment).

In this work, an approach has been developed to segment 2D CT image slices of the bladder using the manual segmentation (drawn by an experienced radiation oncologist) on an adjacent slice as prior knowledge. Our goal is to design a deep network architecture able to capture both the bladder appearance statistics and the deformation occurring between adjacent slices. In that sense, it allows to *propagate* the segmentation between adjacent slices. To demonstrate the relevance of our approach, we analyze quantitatively how the segmentation result improves with the proximity of the adjacent slice. Note that slices segmentation of the bladder is not a trivial task due to the high shape and size variation of the bladder, the presence or not of contrast material in it and the poor contrast between its wall and surrounding soft tissues.

Section 4.2 introduces the CT images that have been considered in our study and their preprocessing. Section 4.3 explains how to turn the segmentation problem into a learning problem, by defining pairs of input/output training samples based on the CT images and their manual annotations by medical doctors. The network architecture is also given in the same section. In Sect. 4.4, the results are provided and discussed. Conclusions are given in Sect. 4.5.

4.2 Materials and Preprocessing

Our data consist of 3D CT volumes (planning CTs, also named PCTs) from patients with prostate cancer. Each patient underwent External Beam Radiation Therapy (EBRT) at CHU-UCL-Namur (198 patients) or at CHU-Charleroi – Hôpital A. Vésale (141 patients). The images coming from both hospitals are shuffled in our dataset and images with and without contrast agent are present in it. For each patient, the bladder was manually delineated in the PCT by an experienced radiation oncologist. This has been considered as the ground truth in this

work. The use of these retrospective, anonymized data for this study as been approved by each hospital ethics committee. The data were pre-processed according to the following steps: data re-sampling, slice selection and cropping and intensity range thresholding.

The 3D CT volumes (stacks of 2D CT slices) extracted in hospitals under DICOM format have a different pixel spacing and slice thickness across different patients. In order to ensure data uniformity over the entire dataset, all the 3D CT volumes have been re-sampled through a linear interpolation according to a 1x1x1 mm regular grid. The bladder contours drawn by the clinicians are stored as 3D binary masks with same sampling as the CT volumes. The DICOM format manipulation and re-sampling are performed using the opensource platform OpenReggui.¹

For every patient, a single 2D axial slice where the bladder is present is randomly selected, together with several adjacent slices, from the 3D CT volume. This is further discussed in Sect. 4.3.1. Every selected slice is cropped to 192x192 pixels tile centered on the bladder. The reason for cropping is that it allows faster experimentation. Our investigations (not described in this paper) have shown that running the algorithm on the entire slice instead of an image tile centered on the region of interest leads to comparable segmentation accuracy.

All pixels intensities above 3000 HU (Hounsfield unit) and below -1000 HU are respectively set to 3000 HU and -1000 HU to avoid artifacts in the high Hounsfield unit range. The lower and upper bounds are respectively determined from the Hounsfield values of air and cortical bone.

4.3 Formulating the Contour Propagation as a Learning Problem

Our goal is to segment target 2D CT image slices of the bladder using the manual delineation on an adjacent slice as prior knowledge. In this section, the target and adjacent slices selection is firstly explained. Then, we present the CNN architecture used for the segmentation and how the prior information is included in it.

¹<https://openreggui.org/>.

4.3.1 Prior Definition and Computation

Each 3D CT volume is made of a stack of 2D slices along the axial axis. For every patient, the bladder only appears on a set of slices indexed by $p \in [0, 1, \dots, P - 1]$ where the bladder top and bottom slices are respectively indexed by $p = 0$ and $p = P - 1$, with P the number of slices containing the bladder. For every patient, a single *target* slice indexed by p_t has been randomly selected such that $p_t \in [0, (P - 1) - 10]$.

Then, for every target slice indexed by $p = p_t$, the corresponding prior knowledge is extracted from an *adjacent* slice indexed by $p_q = p_t + q$ with $q \in [0, 1, \dots, 10]$. Hence, the indices p_q of the prior slices are always such that $p_q \in [0, (P - 1)]$. The prior knowledge used in this work is the manual delineation (contour) drawn by the clinician on the q th adjacent slice. This contour is represented as a binary mask $\mathbf{Y}^{(k)[q]} \in \{0, 1\}^{M \times M}$ defined as

$$Y_{m,n}^{(k)[q]} = \begin{cases} 1 & \text{if } (m, n) \text{ is in the bladder} \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

where (m, n) is the pixel position within the binary mask $\mathbf{Y}^{(k)[q]}$, k is the index of the binary mask in the dataset and M is the image tile width.

4.3.2 Network Architecture and Learning Strategy

The u-net fully convolutional neural network has been used for the bladder segmentation. The architecture of the network is shown in Fig. 4.1 with the hyper-parameters used in this work. Different network architectures and different hyper-parameters have been tested, and the one yielding highest performance on validation set have been selected. The network input is expanded from one to two channels as in [13]. The additional channel is used to enter the prior knowledge into the network.

More precisely, let us assume that the network input for the k th data sample is denoted by $\mathbf{X}^{(k)} \in \mathbb{R}^{M \times M \times C}$, where C is the number of input channels. The corresponding labels are a binary mask $\mathbf{Y}^{(k)} \in \{0, 1\}^{M \times M}$ equal to one within the bladder and equal to zero everywhere else. The notation $\mathbf{X}_{:, :, c}^{(k)}$ is used in order to denote the c th network input channel. The input is the concatenation of the target image $\mathbf{I}^{(k)} \in \mathbb{R}^{M \times M}$ with

the prior mask given on the q th adjacent slice above the target slice: $\mathbf{X}_{::,1}^{(k)} = \mathbf{I}^{(k)}$ and $\mathbf{X}_{::,2}^{(k)} = \mathbf{Y}^{(k)[q]}$ as shown in Fig. 4.2. Eleven different networks are trained, each for a different distance between the target slice and the adjacent slice. Formally, for $q \in [0, 1, \dots, 10]$, eleven different models $\mathcal{M}_q$ are trained, each of them with a training set $\mathcal{S}_{train}^{[q]}$ given by

$$\mathcal{S}_{train}^{[q]} = \left\{ \left(\mathbf{X}^{(j)}, \mathbf{Y}^{(j)} \right) \mid j \in [0, 1, \dots, J_{train} - 1], \mathbf{X}_{::,1}^{(j)} = \mathbf{I}^{(j)}, \right. \\ \left. \text{and } \mathbf{X}_{::,2}^{(j)} = \mathbf{Y}^{(j)[q]} \right\},$$

where J_{train} is the number of target tiles in the training set.

The network architecture, inspired from [143], is illustrated in Fig. 4.1. It takes as input either one channel (if the CNN is used without prior knowledge, see Sect. 4.4.1) or two channels (if the CNN is used with prior knowledge) and outputs a prediction for the target tile bladder segmentation. The input goes through a contracting path (left side) to capture context and an expanding path (right side) to enable precise localization.

In the contracting path, a collection of hierarchical features are learned thanks to successive convolutions and max-pooling operations. Successively, two series of 3x3 convolutions, followed by a ReLu activation and batch normalization are applied before applying a max-pooling 2x2. After each max-pooling step, the number of feature maps is doubled in order to allow the network to learn many high level features. From the features learned in the contracting path, the expanding path increases the resolution via successive 2x2 transposed convolutions (i.e. a 2x2 up-sampling followed by a 2x2 convolution), 3x3 convolutions and ReLu activations in order to recover the original tile size.

In the last layer, a sigmoid is applied and the network outputs the probability for each pixel to belong to the bladder. To obtain the final segmentation, a threshold of 0.5 is chosen. Hence, every pixels with a score above the threshold is predicted to belong to the bladder, while other pixels are predicted not to belong to the bladder. We performed experiments (not described in this article) showing that choosing different threshold did not yield significantly superior performance.

The network is trained with the Dice loss. For the j th training example, the Dice loss between the predicted segmentation $\hat{\mathbf{Y}}^{(j)} \in [0, 1]^{M \times M}$

and the reference segmentation $\mathbf{Y}^{(j)} \in \{0, 1\}^{M \times M}$ is given by

$$\mathcal{L}(\hat{\mathbf{Y}}^{(j)}, \mathbf{Y}^{(j)}) = -\frac{2 \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} Y_{m,n}^{(j)} \hat{Y}_{m,n}^{(j)}}{\sum_{m=0}^{M-1} \sum_{n=0}^{M-1} (Y_{m,n}^{(j)} + \hat{Y}_{m,n}^{(j)})}. \quad (4.2)$$

The fact that each training tile has at least one pixel belonging to the bladder ensures that the denominator is non-zero. This loss drives the output segmentation to have a large overlap with the ground truth segmentation. This network is implemented in Keras [29], with TensorFlow [5] back-end. The optimization algorithm used is Adam with a learning rate of 10^{-4} , a batch size of 20 and the model is trained for 200 epochs. The other parameters (such as initialization strategy and learning rate decay) are left to Keras default. All the tiles in the dataset are shifted and scaled by the mean and the standard deviation of the tiles intensities over the training set. We use the validation set to earlystop the training. Training data is augmented using rotation, shift, shear, zoom and horizontal flip. Although horizontal flip does not yield plausible images in terms of the general anatomy, they have been shown to improve performance. One possible explanation is that the left and right bladder wall are similar, and thus augmenting the training set in this manner allows to learn useful features.

The target slices and their set of adjacent slices are randomly assigned to the training set (179 patients), the validation set (80 patients) and the test set (80 patients). Note that the patient distribution across the training, validation and test sets is the same for the eleven different models.

4.4 Results and Discussion

In this section, the validation metric and the comparison baselines are first presented. Then, our results are discussed.

4.4.1 Validation Metrics and Comparison Baselines

In order to evaluate our results, we use three metrics: the Dice similarity coefficient (DSC) and the Jaccard index (JI) measure the overlap between two binary masks, while the Hausdorff distance (HD) assesses the distance between the contours extracted from those binary masks.

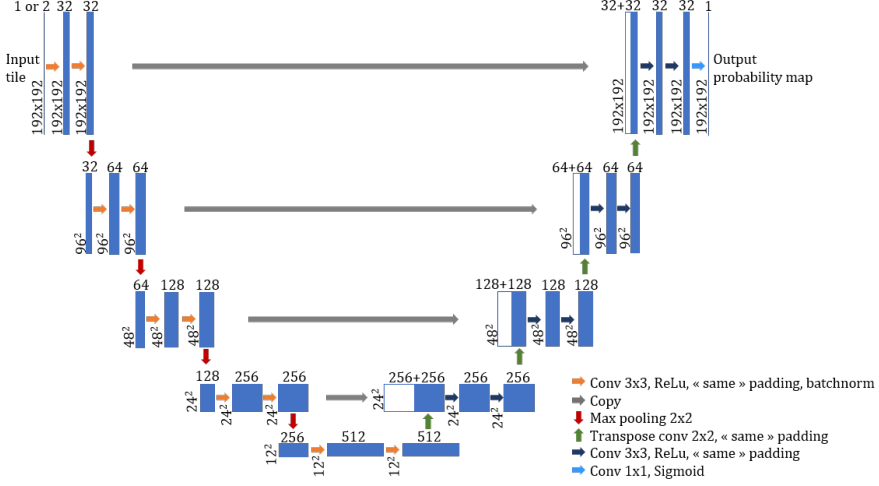


Figure 4.1: The network architecture used is the u-net. Each blue rectangle represents the feature maps resulting from a convolution operation, while white rectangles represent copied featured maps. For the convolutions, the zero padding is chosen such that the image size is preserved ("same" padding). This figure is adapted from [143].

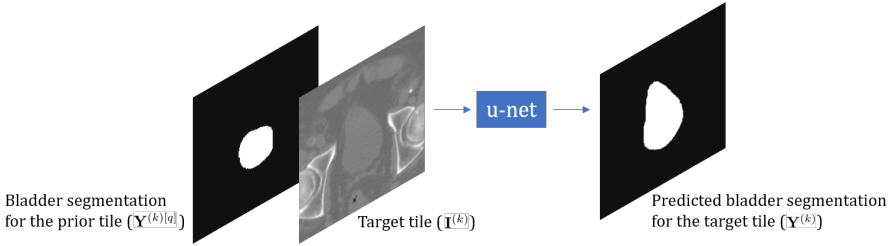


Figure 4.2: The proposed u-net network takes as input the bladder segmentation for the prior tile and the target tile intensities. It outputs a prediction of the bladder segmentation for the target tile.

The DSC and the JI both reach one in case of perfect overlap and zero if there is no overlap between both binary masks. The DSC is slightly different from the Dice loss used for the network training. Indeed, the

DSC is computed between two binary images, the ground truth and the thresholded network prediction, whereas the Dice loss was computed between the ground truth and the network prediction to allow the backward propagation. The JI is defined as the intersection of both binary images over their union. The HD is computed between the ground truth contour $\mathcal{C}_1$ and the prediction contour $\mathcal{C}_2$. Contours are the set of points located at the boundary of the binary masks. More precisely, the HD is the greatest of all the distances from a point in $\mathcal{C}_1$ to the closest point in $\mathcal{C}_2$.

The prediction performed by our network is compared to three different baselines. The *copy* baseline uses the binary mask $\mathbf{Y}^{(k)[q]}$ as a valid approximation of $\mathbf{Y}^{(k)}$. Since the binary mask $\mathbf{Y}^{(k)[q]}$ is given as an input to the network, our algorithm should ideally outperform this baseline.

The u-net network has also been trained without prior knowledge. The network input is simply the target image and has a single channel such that $\mathbf{X}_{::,1}^{(k)} = \mathbf{I}^{(k)}$. This is shown in Fig. 4.1 and denoted as the *CNN without prior* baseline.

The results obtained with our CNN have been compared to the classical registration-based contour propagation. It has been denoted as the *registration* baseline. Such an approach computes a deformation field from the target image $\mathbf{I}^{(k)}$ (fixed image) to the image on the q th adjacent slice $\mathbf{I}^{(k)[q]}$ (the moving image). Applying the inverse of this deformation field to the moving image allows to align the moving image with the fixed image. In turn, applying the inverse of the deformation field to the binary mask on the adjacent slice $\mathbf{Y}^{(k)[q]}$ yields an estimation of the target binary mask. The registration has been performed using SimpleElastix.² The registration components have been chosen according to the recommendations of the Elastix’s user manual [86]. In particular, the mutual information metric, the B-splines transformation and the adaptive stochastic gradient descent optimizer have been used together with a multi-resolution strategy.

²<http://simpleelastix.github.io>.

4.4.2 Discussion

In Fig. 4.3, the DSC between the output segmentation of the CNN with prior and the ground truth segmentation is computed and averaged over the test set for several distances (i.e. from 1 mm to 10 mm) between the target and adjacent slices. This distance is named the inter-slices distance below. The average DSC obtained with the CNN with prior is also compared to the results obtained with the three aforementioned baselines. Table 4.1 provides the mean and standard deviation of the DSC for all the considered methods and an inter-slices distance equal to 2, 4 and 8 mm. The following observations can be done based on Fig. 4.3 and Table 4.1.

- For a large inter-slices distance (i.e. $8 \leq q$), the CNN with prior does not clearly outperform the CNN without prior. This means that the CNN with prior struggles to improve the target segmentation when the prior is not sufficiently correlated with the target segmentation. However, the CNN with prior clearly outperforms the result of the registration-based approach in this region. This is illustrated in the first column of Fig. 4.4.
- For an intermediate inter-slices distance (i.e. $4 \leq q < 8$), the CNN with prior slightly outperforms the CNN without prior. However, the DSC of the copy is still low in this interval. This means that the CNN with prior is able to exploit jointly the information present in the target image and the prior mask. Furthermore, the prior exploitation increases on average as the prior mask becomes more relevant (i.e. for decreasing q). This is illustrated in the second column of Fig. 4.4.
- For a small inter-slices distance (i.e. $q < 4$), the CNN with prior performs like the registration-based approach, which is known to work efficiently when the moving (adjacent) and the fixed (target) images are close to each other. This means that the CNN with prior is able to capture information in the prior mask when it is relevant for the segmentation task. This is illustrated in the third column of Fig. 4.4.

From those observations, we can conclude that the CNN with prior reaches or exceeds the segmentation performance of alternative approaches

for all the considered inter-slices distances. Furthermore, the prior knowledge is more intensively exploited as its correlation with the ground truth increases.

From Table 4.1, it appears that the variance of the considered metrics is relatively high compared to the difference in their mean computed with the different baselines. A finer analysis of the statistical distribution of our segmentation results reveals that, even if the average performance is only moderately better with prior than without prior, the exploitation of the segmentation mask in an adjacent slice (i.e. the prior) prevents a complete failure of the segmentation in worst cases. Indeed, the Dice similarity coefficient is almost always larger than 0.8 in presence of prior, while being lower than 0.8 for about 15 % of the images in absence of prior, with worst cases below 0.6 and reaching 0.2. Hence, we conclude that our proposed framework has a limited impact on the mean Dice performance but significantly increases the robustness of the approach, which is certainly important in practice (e.g. regarding organ segmentation for radiotherapy treatment).

Figure 4.3 and Table 4.1 also include the segmentation performance of a single CNN (in opposition to the eleven CNNs) trained over all the training samples, hence corresponding to different inter-slices distances. This CNN is named *CNN with prior single*. This unique CNN is then tested on the different inter-slices separately. It appears that this network improves the segmentation performance with respect to the CNN without prior for all the considered inter-slices distances, even if it is outperformed by the CNNs trained for a specific inter-slide distance for $q < 4$. This result is encouraging for applications where we cannot define a proper inter-slices distance (e.g. regarding the propagation of a contour over a temporal sequence of images).

The training time was respectively 446 s and 488 s for the CNN without prior and for the CNN with prior. This training time is an average over the training times obtained for the three inter-slices distances considered in Table 4.1. The inference time was respectively ~ 0 s, 3.58 s, 0.057 s and 0.184 s for the copy, the registration, the CNN without prior and the CNN with prior. This inference time is the time necessary to predict the segmentation of a single image, averaged over the three inter-slices distances and over all the test images. It appears that the CNN with and without prior have comparable training times. It can be also

Table 4.1: Comparison of the different segmentation algorithms regarding to their DSC and JI on test set. See text for details. DSC: Dice similarity coefficient. JI: Jaccard index. CNN wop: CNN without prior, CNN wp: CNN with prior (eleven different networks are trained, each one for a given inter-slices distance), CNN wps : CNN with prior single (one single network is trained on all different inter-slices distances).

Algorithm	$p_t - 8$		$p_t - 4$		$p_t - 2$	
	DSC	JI	DSC	JI	DSC	JI
Copy	.761 $\pm$.176	.645 $\pm$.214	.866 $\pm$.122	.781 $\pm$.166	.924 $\pm$.077	.867 $\pm$.120
Registration	.858 $\pm$.147	.775 $\pm$.186	.920 $\pm$.101	.864 $\pm$.130	.948 $\pm$.048	.905 $\pm$.080
CNN wop	.897 $\pm$.159	.840 $\pm$.191	.897 $\pm$.159	.840 $\pm$.191	.897 $\pm$.159	.840 $\pm$.191
CNN wp	.915 $\pm$.103	.856 $\pm$.140	.935$\pm$.071	.885$\pm$.104	.951$\pm$.036	.908$\pm$.062
CNN wps	.919$\pm$.088	.861$\pm$.128	.935$\pm$.055	.883 $\pm$.088	.929 $\pm$.063	.873 $\pm$.099

observed that the inference time of CNNs is one to two orders of magnitude below the inference time of the registration-based method. The CNNs have been trained and tested with nVidia cuDNN acceleration on a computer equipped with a GPU nVidia GeForce GTX 1080 Ti 11Gb.³

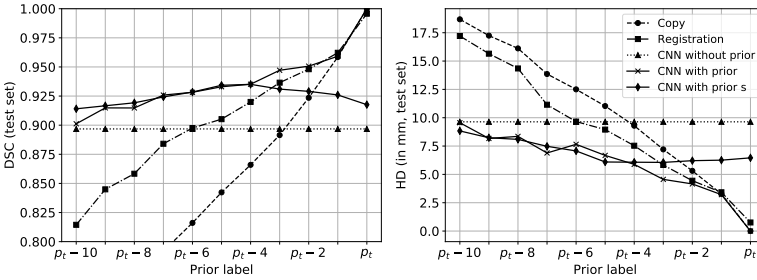


Figure 4.3: Influence of the prior on the DSC and on the HD for test set examples. DSC: Dice similarity coefficient. HD: Hausdorff distance.

³<https://developer.nvidia.com/cudnn>.

4.5 Conclusion

We propose a deep CNN network able to segment 2D CT image slices of the bladder, using the manual delineation on an adjacent slice as prior knowledge. This is done by training the u-net network with two input channels. The first channel is used to enter the target image and the second one is used to enter the manual delineation of the adjacent slice given as a binary mask.

The network is trained and tested for an increasing distance between the target and adjacent slices (i.e. from 1 to 10 mm). The Dice similarity coefficient between the thresholded network prediction and the ground truth exceeds 0.9 for all the considered distances and increases as the distance between the slices decreases.

For every distance, the output segmentation outperforms in average the segmentations obtained by a registration-based approach (efficient for small deformations between both slices) and by the u-net network trained and tested without prior knowledge (efficient for large deformations between slices).

In the future, we plan to use the proposed approach on other organs and other applicative contexts where a spatial or temporal contour propagation is needed. We also plan to investigate how the annotation of a few slices within a 3D volume can improve the segmentation performance of a 3D CNN such as 3D u-net [31]. Our code will also be made available at the time of the conference.

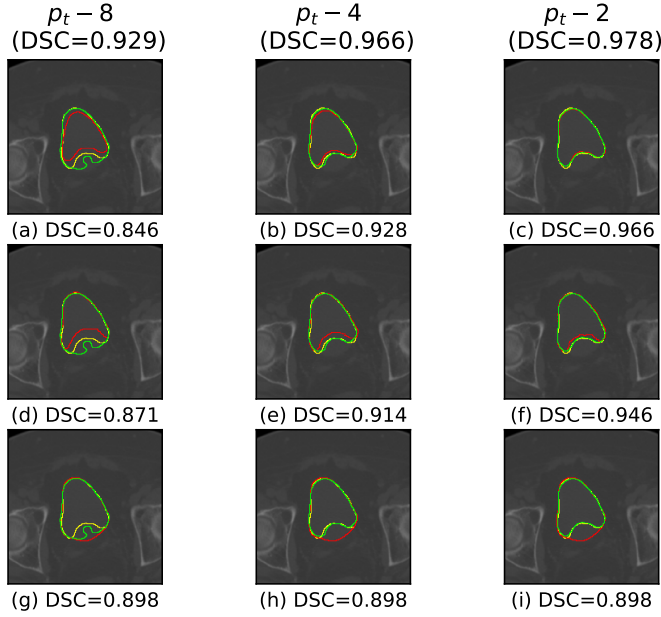


Figure 4.4: Ground truth and predicted contours for a given patient at different inter-slices distances. This figure compares the CNN with prior (in green) with the ground truth (in yellow) and with the baselines (in red, each line showing a different baseline algorithm). In the first row, the baseline is the "copy" algorithm, with the DSC between the ground truth and this baseline reported in (a), (b) and (c). In the second line, the baseline is the registration algorithm. In the third row, the baseline is the CNN without prior. The DSC between brackets are computed between the ground truth and the CNN with prior. The first, second and third columns respectively correspond to an inter-slices distance of 8, 4 and 2 mm. DSC: Dice similarity coefficient.

Chapter 5

Using planning CTs to enhance CNN-based bladder segmentation on cone beam CT

In the previous chapter, we proposed a semi-automated method based on knowledge propagation to contour structures of interest on the planning CT, thereby saving time for medical doctors. While semi-automated methods are adequate for planning, where a doctor is available, they are most often not suited for contouring daily images, which are numerous and for which the doctor does not have to intervene. Moreover, training deep learning models for daily image contouring is challenging since those images (called Cone Beam CTs) are not contoured in clinical routine. In the publication [18], on which this chapter is based, we propose to share the knowledge present in the database of contoured CTs and apply this knowledge to fully-automated segmentation of CBCTs.

5.1 Introduction

External beam radiation therapy (EBRT) treats cancer by delivering daily fractions of radiation to a tumor volume while attempting to spare normal tissues. Current techniques allow for planning and delivery of

complex dose distributions in order to improve dose delivery in the target volume while better sparing surrounding healthy organs. At treatment planning, clinicians delineate the tumor and organs volumes on a computed tomography (CT) and compute the dose distribution. At treatment delivery, the patient is aligned with its treatment planning position and the dose fraction is delivered. Patient positioning relies on a daily cone beam computed tomography (CBCT) volume acquired in treatment position before each treatment fraction. Importantly, large day-to-day variations occur in the pelvic region due to matter intake and outcome (e.g. bladder filling) and can impair treatment dose conformity [134]. Hence, anatomical variations detection (between planning and treatment stages) on the CBCT volume is needed. Current clinical practice includes it through a CBCT visual inspection but the automatic pelvic organs segmentation in daily CBCT volumes would allow an accurate measure of the anatomical variations and better prevent non-compliant dose delivery. The automation of the segmentation is required for an integration in the clinical workflow since manual organs delineation on daily volumes is too time-consuming. Our proposed approach focusses on the automatic bladder, rectum and prostate segmentation on CBCTs with u-net, a fully convolutional neural network. Since annotations are hard to collect for CBCT volumes, we consider augmenting the training dataset with annotated CT volumes following a supervised domain adaptation strategy.

Automatic organs segmentation in CBCT volumes is challenging due to poor soft tissue contrast and the various reconstruction artifacts [165]. Two main approaches have been previously proposed to tackle this. Classical clinically used segmentation methods include deformable image registration (DIR) methods between planning CT and daily CBCT volumes [153, 156, 168] to segment the bladder in CBCTs. Although it shows improvement compared to rigid registration, a large proportion of the obtained bladder contours are unacceptable in case of large deformations between the registered volumes as in the pelvic region [174]. Statistical shape models [67] allow to capture shape variations and have been used for bladder segmentation on CBCT [24, 162]. However, those methods require the definition of landmarks or meshes. Also, building a patient specific shape model [24] requires the availability of several delineated CBCT volumes, which is hampering the application from the

start of the treatment. Alternatively, in order to limit the number of required delineated CBCT volumes while following the shape variation along a treatment, the model is best updated on segmentations manually corrected during the treatment [162]. However, this requires user intervention. Henceforth, none of these methods closes the challenging task of pelvic organs segmentation on CBCT volumes.

In parallel, the recent advances in computing capabilities, the availability of representative datasets and the high versatility of deep learning approaches allowed them to reach impressive segmentation performance, competing with state-of-the-art segmentation tools. In contrast with the above-mentioned segmentation techniques, deep learning algorithms are supposed to be robust to shape and appearance variations if those variations are captured in the training database and they do not require a landmarks definition. Deep learning algorithms have already been successfully used to segment pelvic organs on CT images, including the urinary bladder [23, 82]. Hence, our goal is to perform fully automatic bladder segmentation in CBCT volumes with u-net, a 3D fully convolutional neural network (FCN) [143]. To our knowledge, there has not been any attempt yet to use deep learning to segment the bladder in CBCTs, probably due to the scarcity of annotated data. In order to deal with of the small amount of annotated daily CBCT volumes, the performance improvement provided by the augmentation of the training dataset with annotated planning CT volumes is proposed. We motivate this choice by the fact that CBCT volumes can be roughly considered as noisy and distorted CT volumes in a segmentation perspective, hence sharing shape and contextual information with the CT volumes. We investigate the performance of u-net for different numbers of training CT and for different numbers of training daily CBCT, and show that adding CTs in the training set helps for contouring CBCTs.

Our approach is an instance of transfer learning and, more precisely, an instance of supervised domain adaptation. *Transfer learning* refers to the situation where a model is trained either (i) using data coming from different domains or (ii) on different tasks or (iii) both using data coming from different domains and on different tasks [133]. Our model is trained on two similar but different domains (CT and CBCT) on the same task (classifying each voxel of a 3D matrix as bladder, rectum, prostate, or none of these). The CT domain is called *source* domain

since it is the one from which most training data come, while the CBCT domain is called *target* domain since it is the one for which we want our model to perform well. As it is the case in most transfer learning applications, our source domain data are used to mitigate the small number of available target data during training. The situation where source and target domains are different is referred as *domain adaptation*, which is a subclass of transfer learning [76]. In our approach, all target data are labeled (i.e., we dispose of a contour of the organs for all CBCTs) and therefore we are in the situation of *supervised domain adaptation*. Our approach shares similarities with Kamnitsas et al. [80], where brain lesions are segmented using datasets coming from different domains. The key difference with our work is that their target domain data are unlabeled (i.e., they perform *unsupervised domain adaptation*). However, they compare their approach with a network jointly trained with labeled source and target data, which is more similar to our approach. Ghafoorian et al. [55] perform supervised domain adaptation to learn brain lesion segmentation on MRI across FLAIR and T1 images. However, they first pre-train the network on the source domain dataset before fine-tuning it on the target domain dataset. We jointly train our network on both source and target data.

In Section 5.2, we present both datasets of labeled CTs and CBCTs used in this study. We also introduce the u-net architecture, our learning strategy and the comparison baselines. In Section 5.3, we report the results and discuss them. Finally, we conclude in Section 5.4.

5.2 Materials and methods

In this section, we propose a method to assess whether adding CTs in the training set boosts the performance of a neural network to segment CBCTs. We use a dataset of 112 CTs and 48 CBCTs, presented in Section 5.2.1. These data are fed to the u-net neural network, whose architecture is described in 5.2.2. In order to evaluate the gain brought by CTs, we evaluate u-net’s performance on a fixed dataset of 48 CBCTs in different training settings (i.e., with a varying number of training CTs and training CBCTs). The cross-validation scheme detailed in Section 5.2.3 improves the statistical relevance of our results, mitigating the

small size of our dataset. The performance metrics are defined in the same section. The performance of u-net in the different training settings is compared to other segmentation algorithms (using a commercial software on our data and reporting the results from other authors on other datasets) in Section 5.2.4.

5.2.1 Data and pre-processing

Our data consist of (i) a set $\mathcal{S}_1$ of 64 patients for which we have a CT which has been delineated by a trained expert and (ii) a set $\mathcal{S}_2$ of 48 patients (different from the 64 patients mentioned above) for which we have a planning CT and a daily CBCT (acquired with a Varian TrueBeam STx version 1.5), which have been both delineated by a trained expert. The patients of sets $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively underwent EBRT at CHU-UCL-Namur and CHU-Charleroi Hôpital André Vésale. The use of these retrospective, anonymized data for this study has been approved by each hospital's ethics committee. In order to ensure data uniformity across the entire dataset, all the 3D CT and CBCT volumes (as well as the 3D binary mask representing the ground truth segmentations) have been re-sampled on a 2x2x2 mm regular grid. Every re-sampled image volume and binary mask volume are cropped to volumes of 96x96x80 voxels centered on the bladder. Downsampling and cropping allow the model parameters, each batch of eight image volumes and the corresponding eight binary mask volumes to fit in a 11 Gb GPU.

5.2.2 Network architecture

The 3D u-net fully convolutional neural network is considered in this study. The network follows the same architecture (i.e. number and composition of layers) as in Ronneberger et al. [143], where the 3x3 convolutions, the 2x2 max-pooling and the 2x2 up-conversion operations have been replaced by their 3x3x3 and 2x2x2 counterparts, as in Çiçek et al. [31]. The network takes as input the 96x96x80 image volume and outputs a prediction for the bladder segmentation. The input goes through a contracting path to capture context and an expanding path to enable precise localization. In the contracting path, a collection of features are learned thanks to successive convolutions and max-pooling operations. Successively, two 3x3x3 convolutions, followed by a ReLu

activation, are applied before applying a $2 \times 2 \times 2$ max-pooling. After each max-pooling step, the number of feature maps is doubled in order to allow the network to learn many high level features. Within a layer, the number of feature maps is kept constant as in Ronneberger et al. [143], starting with 16 feature maps in the first layer. From the features learned in the contracting path, the expanding path increases the resolution via skip connections, successive $2 \times 2 \times 2$ up-conversion operations, $3 \times 3 \times 3$ convolutions and ReLu activations in order to recover the original volume size. In the last layer, a sigmoid is applied and the network outputs the probability for each voxel to belong to the bladder. To obtain the final binary segmentation mask, a threshold of 0.5 is chosen. The network is trained with the Dice loss. The optimization algorithm used is Adam with learning rate 10^{-4} and a batch size of 8. We use the validation set to earlystop the training (with a maximum of 100 epochs). Training data (both CBCT and CT image volumes as well as their binary mask volumes) are augmented using rotation (comprised between -5° and 5° along each of the three axes), shift (comprised between -5 and 5 pixels along each axes) and shear (reasonable values for the affine transformation matrix).

5.2.3 Learning strategy and performance assessment

In the rest of this paper, a "CT volume" (or "CBCT volume") refers to a couple of both the 3D image and the 3D binary mask representing the bladder segmentation on this image. We perform a six-fold cross-validation (see Table 6.1) with the 48 CBCT volumes of set $\mathcal{S}_2$, where four folds ($n_{CBCT} \leq 32$ volumes in total) are used as training set, one fold (8 volumes) is used as validation set for earlystopping and one fold (8 volumes) is used as test set. As shown in Figure 5.1, the number of training CBCT volumes n_{CBCT} has been varied such that $n_{CBCT} \in \{2, 4, 8, 16, 32\}$ by excluding a part of the 32 available volumes. The training set has been augmented with n_{CT} annotated CT volumes from set $\mathcal{S}_1$ such that $n_{CT} \in \{0, 16, 32, 64\}$. The same CT volumes are added to the CBCT training volumes independently on the four considered training folds. Hence, the training set contains $n_{CBCT} + n_{CT}$ volumes in total. Note that no CT volumes are present in the validation and test sets (indeed our goal is only to segment CBCT volumes). In order to evaluate our results, we use three metrics: the Dice similarity coefficient

Table 5.1: Six-fold cross-validation. To train the model, we use n_{CT} volumes from $\mathcal{S}_1$ and we use the n_{CBCT} first volumes from the CBCT folds labeled with "train". To earlystop the training, we use all eight volumes from the CBCT fold labeled with "val". To test the model, we use all eight volumes from the CBCT fold labeled with "test".

$\mathcal{S}_1$	$\mathcal{S}_2$ (CBCT)					
(CT)	fold1	fold2	fold3	fold4	fold5	fold6
train	train	train	train	train	<i>val</i>	test
train	train	train	train	<i>val</i>	test	train
train	train	train	<i>val</i>	test	train	train
train	train	<i>val</i>	test	train	train	train
train	<i>val</i>	test	train	train	train	train
train	test	train	train	train	train	<i>val</i>

(DSC) and the Jaccard index (JI) measure the overlap between two binary masks, while the symmetric mean boundary distance (SMBD) assesses the distance between the contours (i.e. the sets of points located at the boundary of the binary masks) extracted from those binary masks. More specifically,

$$\text{DSC} = \frac{2|A \cap B|}{|A| + |B|}, \quad (5.1)$$

$$\text{JI} = \frac{|A \cap B|}{|A \cup B|}, \quad (5.2)$$

$$\text{SMBD} = \frac{\overline{D}(A, B) + \overline{D}(B, A)}{2}, \quad (5.3)$$

where A and B are the predicted and reference segmentation binary masks, $D(A, B) = \{\min_{x \in \Omega_B} \|x - y\|, y \in \Omega_A\}$ and Ω_A, Ω_B are respectively the contours extracted from A and B .

5.2.4 Comparison baselines

The prediction performed by our network is compared to several baselines. A first baseline has been obtained on the dataset presented in Sec-

tion 5.2.1 by using the built-in contour propagation tool of RaySearch (RayStation¹ version 5.99.50.22). The contours from the planning CT volumes of set $\mathcal{S}_2$ have been propagated to the daily CBCT image volumes of the same patient by using a rigid registration followed by a regularized intensity-based deformable image registration (DIR). More precisely, the DIR algorithm is the ANACONDA algorithm [166], where no controlling regions of interest (ROI) have been used, apart from the external/whole body structure. This follows the approach proposed in Takayama et al. [153] when no controlling ROIs (i.e. annotations) are available on the CBCT image volumes for the bladder (or other organs).

Additional baselines are reported on others CT-CBCT datasets obtained with different acquisition machines. A rigid registration followed by an intensity-based DIR between pairs of planning CT image volumes and daily CBCT image volumes is performed in Thor et al. (demon algorithm) [156] and Takayama et al. (ANACONDA algorithm without controlling ROIs) [153]. In Woerner et al. [168], several mutual information-based DIRs are performed after the rigid registration with a user intervention at the end of the process in order to tune a final DIR.

A patient specific bladder deformation model is proposed in Chai et al. [24] and van de Schoot et al. [162]. In Chai et al. [24], five delineated CBCT volumes serve as training data in order to build a statistical bladder shape model through principal component analysis (PCA) on shape vectors capturing the 3D position of 2091 landmarks on the bladder contour. Then, the test volume segmentation is obtained by deforming a reference segmentation (namely one of the training contours) along the major deformation modes obtained with PCA. This is done by the minimization of the absolute difference between the directional intensity gradients evaluated on the bladder contour in the reference CBCT and the bladder contour candidate in the test CBCT. In van de Schoot et al. [162], the statistical shape model is built on two planning CT volumes instead of five treatment CBCT and updated on manually corrected contours during the treatment. The absolute difference minimization is also replaced by a cross-correlation maximization.

¹<https://www.raysearchlabs.com/raystation/>.

5.3 Results and discussion

In Figure 5.1, the DSC between the output segmentation of the FCN and the ground truth segmentation are computed and averaged over all 48 CBCT volumes from the six test folds. This is done for different numbers of training CBCTs and different numbers of training CTs. Those results are compared with the DIR algorithm of RayStation. Table 6.3 provides the mean and the standard deviation of the DSC, JI and SMBD for different numbers of training CBCT volumes and different numbers of training CT volumes. A mixed model with a random intercept on the patient showed significant difference between the algorithms' performance regarding their DSC ($p < 10^{-3}$), JI ($p < 10^{-3}$) and SMBD ($p < 10^{-3}$). The following three observations can be done based on Figure 5.1 and Table 6.3.

- The more CBCT volumes in the training set, the higher the DSC on the test set. A Tukey's range test performed on DSC reveals that Ours(32,0) performs significantly better than Ours(8,0) ($p < 10^{-3}$) and that Ours(32,64) performs significantly better than Ours(8,64) ($p < 10^{-3}$). This means that adding new CBCT volumes allows the network to better generalize on test set. This makes sense since the added training data and the test data are of the same modality (CBCT).
- Interestingly, the more CT volumes in the training set, the higher the mean DSC (Ours(8,64) > Ours(8,0) with $p < 10^{-3}$ and Ours(32,64) > Ours(32,0) with $p < 10^{-2}$). We explain this improvement by the learning of more generic features, leading to better generalization. However, the more training CBCT volumes, the smaller the gain obtained from additional CTs in the training set.
- When all available CT and CBCT volumes are used for training (32 CBCT volumes and 64 CT volumes), our approach outperforms the DIR algorithm of RayStation (Ours(8,64) > RayStation with $p < 10^{-3}$). This is illustrated on a representative patient in Figure 5.2. Our approach also outperforms the DSC metrics presented in Thor et al. on a different but comparable dataset. Note that those two DIR-based comparison baselines perform similarly, which suggests that our dataset is of similar difficulty than the one

used by Thor et al. Our method also outperforms the results of Takayama et al. when their algorithm is used without controlling ROIs. Our method reaches results comparable to Woerner et al. but does not require a user intervention and the prediction requires a single pass in the CNN compared to a 6-pass deformable registration in Woerner et al. Although our algorithm performs worse than the one proposed by Chai et al. and van de Schoot et al., it is not patient specific. Hence, it does not require the availability of patient specific CBCT (or CT) contours beforehand, nor a model update during the treatment.

Although this discussion is based on the DSC, basing it on the JI or on the SMBD leads to the same observations with the same levels of confidence. From those observations, we conclude that our FCN outperforms alternative fully-automatic approaches and is slightly worse than a semi-automatic approach. Furthermore, adding in the training set volumes from a distribution (CT) different but close to the distribution of the test set (CBCT) improves the FCN performance.

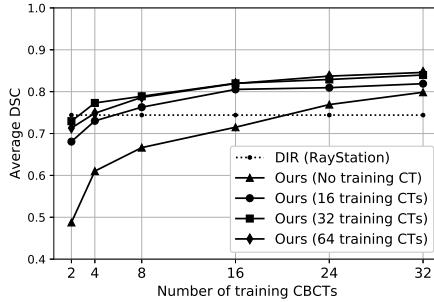


Figure 5.1: Influence of the number of the training CBCTs and CTs on the segmentation accuracy. DSC: Dice similarity coefficient.

5.4 Conclusion

The contribution of this paper is twofold. (i) To the best of our knowledge, this is the first attempt to use convolutional neural networks to

Algorithm (n_{CBCT}, n_{CT})	DSC	JI	SMBD (mm)
Ours (8,0)	.669 $\pm$.155	.521 $\pm$.159	6.5 $\pm$ 3.1
Ours (8,64)	.788 $\pm$.110	.663 $\pm$.136	3.9 $\pm$ 1.9
Ours (32,0)	.801 $\pm$.137	.685 $\pm$.147	3.9 $\pm$ 2.6
Ours (32,64)	.848 $\pm$.085	.745 $\pm$.114	2.8 $\pm$ 1.4
DIR, RayStation	.744 $\pm$.144	.612 $\pm$.167	5.0 $\pm$ 3.1
DIR, Takayama et al. (2017) [153]*	.69 $\pm$.07	-	-
DIR, Woerner et al. (2017) [168]*	$\sim$.83	-	-
PSM, van de Schoot et al. (2014) [162]*	$\sim$.87	-	-
PSM, Chai et al. (2012) [24]*	-	.785	-
DIR, Thor et al. (2011) [156]*	.73	-	-

Table 5.2: Comparison between our proposed algorithm in different settings (number of training CBCT volumes, number of training CT volumes) and the baseline algorithms regarding to overlap (DSC, JI) and distance (SMBD) measures. DSC: Dice similarity coefficient, JI: Jaccard index, SMBD: Symmetric mean boundary distance, DIR: Deformable image registration, PSM: Patient specific model. *Evaluated on a dataset different from ours.

segment the bladder on CBCT. (ii) We show that including in the training set volumes from a distribution (CT) close but different from the test set distribution (CBCT) improves the performance on this test set for the task at hand. Our proposed learning strategy leverages largely available labeled CT volumes available from radiotherapy planning to outperform current state-of-the-art DIR-based bladder segmentation on CBCT.

Convolutional neural networks result in accurate segmentation of the bladder for CBCT volumes. Moreover, we demonstrate that augmenting the training set with CT volumes (which are more accessible since labeling them is part of clinical practice, in contrary to CBCTs) improves the segmentation performance. Our FCN could be used in radiotherapy to precisely localize the bladder, allowing to reduce the dose delivered to this healthy organ in patients treated for prostate cancer. In future works, we could attempt to explain how deep features of CBCTs are improved by CT training data. Further analysis could also investigate how the information present in the annotated planning CT of a given patient can be leveraged to segment the subsequent CBCTs of the same

patient.

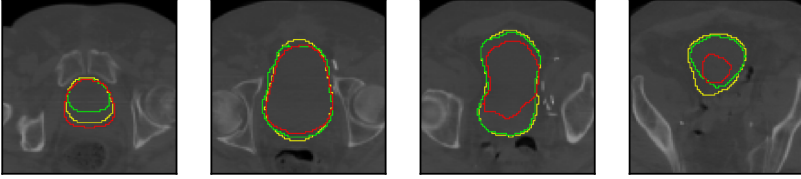


Figure 5.2: Comparison between the ground truth segmentation (in yellow), the DIR with RayStation segmentation (in red, $DSC = 0.788$) and our segmentation (in green, $DSC = 0.892$, setting $n_{CBCT} = 32$ and $n_{CT} = 64$) for a given patient. Each image represents a different slice of the same CT volume. DIR: Deformable image registration.

Chapter 6

Cross-domain data augmentation for deep-learning-based male pelvic organ segmentation in cone beam CT

In the previous chapter, we proposed a method using knowledge transfer from CT to CBCT to automatically contour the bladder, an organ at risk, on Cone Beam CT. However, adaptive radiotherapy requires the monitoring of another organ at risk, the rectum, as well as the target, which often corresponds to the whole prostate. In this chapter, we extend the method from the previous one to propose a method for segmenting the bladder, rectum, and prostate.

The resolution of the images is coarser in this chapter than in the previous one ($1.2 \times 1.2 \times 1.5$ mm, versus $1 \times 1 \times 1$ mm). Coarser resolutions lead to smaller images in matricial representation. This had the advantage of speeding up the numerous, lengthy simulations of this chapter. We assume that it did not impact our conclusions. Verifying this claim is left for future work.

Surprisingly, we observe that a network trained with only 20 CBCTs already yields reasonable performance. This seems to be in contradic-

tion with the hundred of thousands or millions of examples that we mentioned to be necessary for natural images in the introductory chapter of the thesis. First, the images that we use here are 3D images (versus 2D images in natural images). One single image thus contains more information than in natural images. Second, let us remind the reader that hundreds of thousands or millions of examples are required to *match* human performance. Here, our network most probably does not match human performance. Third, the task is different. It may be that reaching human-level performance contouring organs is easier than the task of natural image classification. Let us emphasize that it does not mean that the contouring task is easy *per se*: studies on inter- and intra-observer variabilities suggest that humans themselves are not especially good at it.

In this work, the network’s parameters were initialized randomly. Another possibility would be to start with the weights from the first half of an encoder-decoder trained on CT scans. This is left for future work.

This chapter assumes that annotations for CBCT are scarce because contouring them is not part of the current clinical routine. Let us mention here that this could change in the future with the launch of new adaptive workflows requiring these contours. See for example Varian’s ETHOS system: <https://www.varian.com/fr/products/adaptive-therapy/ethos>.

To validate Schreier et al.’s [145] hypothesis that higher performance is due at least in part to the fact that their test set includes a few CTs, one could add CTs to our test set as well and investigate whether that raises the average test performance. This is left for future work.

A reference to the optimization algorithm used (Adam) has been added to the original article [101].

6.1 Introduction

Fractionated external beam radiotherapy (EBRT) cancer treatment relies on two steps. In the treatment planning phase, clinicians delineate the tumor and surrounding healthy organs’ volumes on a computed tomography (CT) scan and compute the dose distribution. In the treatment delivery phase, the patient is aligned with a specific treatment

planning position, and the dose fraction is delivered. Patient positioning relies on a daily cone beam computed tomography (CBCT) scan acquired in the treatment position before each treatment fraction is delivered.

CT and CBCT are both based on X-ray propagation through the patient's body. However, the CBCT scans are of lower quality than CT scans due to different types of artifacts, including noise, beam hardening, and scattering, as shown in Figure 7.1. In particular, scattering is an important limitation that could rule out the use of CBCT for radiotherapy treatment planning [20]. However, CBCT scans are currently used in order to detect daily variations in patient anatomy, which are particularly large in the pelvic region due to physiological function (e.g., bladder and rectal filling and voiding). Detecting such variations is important since they can impair treatment dose conformity, which means delivering too large a dose to the healthy organs (e.g., the bladder and rectum in the case of prostate cancer) and too low a dose to the clinical target volume (which simply corresponds to prostate itself for a significant proportion of patients) [134]. To improve treatment dose conformity in the pelvic region further, proposals have been made to change treatment plan delivery as a function of time based on observed anatomic variations [56, 137].

However, a step towards better adaptive radiotherapy would require automatic segmentation of the pelvic organs on daily CBCT scans in order to measure the anatomical variations accurately. Automating this segmentation is necessary to be able to integrate it in the clinical workflow, as delineating the organs manually on daily scans is excessively time consuming. Measuring anatomical variations is particularly important in proton therapy because the proton dose distribution is highly sensitive to changes in patient geometry [122, 164].

Currently, organ segmentation is classically performed by deformable image registration (DIR) algorithms between the planning CT and daily CBCT scans [129, 142]. These algorithms include such clinical software packages as MIM [123] and RayStation [153]. Although the results are better than those of rigid registration, these intensity-based DIR algorithms fail in the presence of large deformations between the registered scans, as is the case in the pelvic region [156, 174]. Zambrano et al. [174] and Thor et al. [156] implemented a featurelet-based algorithm [150] and

the demons DIR algorithm [155], respectively. As a result, more complex DIR approaches, such as a B-spline DIR algorithm relying on mutual information, have been proposed [168]. This last approach implements a six-pass DIR with progressively finer resolution and, after visual inspection, an optional final pass using a narrow region around the region of interest. Another approach uses a DIR framework where a locally rigid deformation is enforced for bone and/or prostate, while surrounding tissue is still allowed to deform elastically [90]. Alternatively, statistical shape models can capture shape variations and have also been considered for bladder segmentation on CBCT scans [24, 162]. However, those methods require the definition of landmarks or meshes. Moreover, several delineated CBCT scans must be available to build a patient-specific shape model. That thwarts the application of such methods at the start of treatment. Therefore, none of these methods accomplishes the challenging task of pelvic organ segmentation on CBCT scans. In parallel, recent advances in computing capabilities, the availability of representative datasets, and the great versatility of deep-learning (DL) approaches have enabled DL algorithms to achieve impressive segmentation performance. Unlike the aforementioned techniques, DL algorithms are supposed to be robust to variations in shape and appearance if those variations are captured in the training database and do not require landmark definition. DL algorithms have already been used successfully to segment pelvic organs on CT scans [23, 82]. The 3D U-net fully convolutional neural network [31] has been used to segment female pelvic organs on CBCT scans [62, 64]. Concurrently, we showed that adding annotated CT scans to the training set improved bladder segmentation on CBCT scans [18]. This approach was motivated by the scarcity of annotated CBCT scans, compared with annotated CT scans, and the fact that CBCT scans can be roughly considered to be noisy, distorted CT scans from a segmentation perspective, hence sharing shape and contextual information with the CT scans. The current paper extends our previous conference paper [18] in that it considers additional male pelvic organs (rectum and prostate) and presents more comparative results (including the morphons' deformable registration algorithm). It also involves data from an additional hospital and provides a more detailed discussion. Segmentation of male pelvic organs (bladder, rectum, prostate, and seminal vesicles) on CBCT and CT scans using a DL approach was

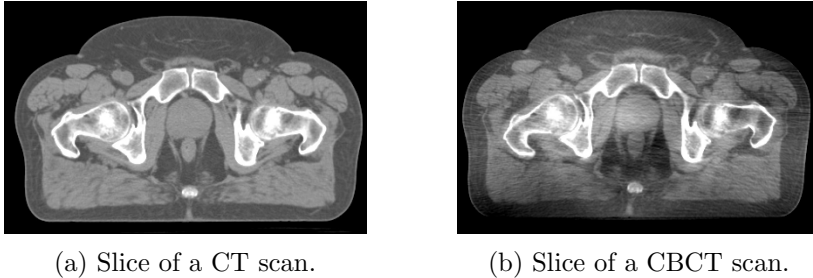


Figure 6.1: Comparison of CT and CBCT scans.

the subject of a recent paper [145]. These authors' contribution consists mainly of the use of artificially-generated pseudo CBCT scans in the training set along with a high segmentation quality. Our approach adds training on real CBCT scans and provides a new and larger test set, as well as more extensive comparison with clinically-used registration tools.

The main contributions of this work are to provide (i) a DL-based segmentation method for male pelvic organs on CBCT scans and (ii) a detailed comparison of state-of-the-art segmentation tools in order to guide the choice of method in clinical practice. The impacts of the number of training scans and the addition of CT scans to the training database are studied in order to provide detailed information on the amount of annotations required for use in clinical practice.

6.2 Materials and Methods

6.2.1 Data and Preprocessing

Our data consisted of (i) a set S_1 of 74 patients for whom we had delineated CT scans and (ii) a set S_2 of 63 patients (different from the 74 patients mentioned above) for whom we had delineated planning CT scans and delineated daily CBCT scans. The contours of bladder, rectum, and prostate were delineated on the CT scans during the clinical workflow. The contours on the CBCT scans were delineated by a trained expert specifically for this study. Within set S_1 , 18 and 56 patients underwent EBRT for prostate cancer at two teaching hospitals, CHU-Charleroi Hôpital André Vésale and CHU-UCL-Namur, respectively. Within set S_2 , 23 and 40 patients underwent EBRT for prostate cancer

at CHU-Charleroi Hôpital André Vésale (CBCT scans acquired with a Varian TrueBeam STx version 1.5) and CHU-UCL-Namur (CBCT scans acquired with a Varian OBI cone beam CT), respectively. The use of these retrospective, anonymized data for this study was approved by each hospital’s ethics committee (dates of approval: 24 May 2017 for CHU-Charleroi Hôpital André Vésale and 12 May 2017 for CHU-UCL-Namur). In order to ensure data uniformity across the entire dataset, all 3D CT and CBCT scans (as well as the 3D binary masks representing the manual segmentations) were re-sampled on a $1.2 \times 1.2 \times 1.5$ mm regular grid. All re-sampled image volumes and binary mask volumes were cropped to volumes of $160 \times 160 \times 128$ voxels containing bladder, rectum, and prostate.

The case selection procedure is described in Figure 6.2. Patients with an artificial hip were excluded from this study because the presence of an artificial hip degraded the image too much for the organs to be segmented accurately by a human expert. Patients for whom prostate was not contoured on the planning CT scan were also excluded. This corresponded to patients for whom the clinical target volume (CTV) differed from that of prostate, either because this organ had been surgically removed or because the CTV included other areas in addition to prostate. Note that it is common in radiotherapy to inject contrast media into bladder. Different inter-subject levels of contrast product increased the variability of this organ’s appearance, making its automatic contouring more challenging. Since our case selection procedure included all patients regardless of the use of contrast media, our method was supposed to be robust to such variability.

6.2.2 Model Architecture and Learning Strategy

Bladder, rectum, and prostate were segmented on CBCT scans using the 3D U-net fully convolutional neural network [31,143]. The 3D input went through a contracting path to capture context and an expanding path to enable precise localization. In the last layer, a softmax was applied, and the network output the probability of each voxel belonging to bladder, rectum, prostate, or none of these organs. The network architecture is shown in Figure 6.3. To obtain a binary mask for each organ, the most probable class label was assigned to each pixel individually. In practice, each organ was segmented as a single region of

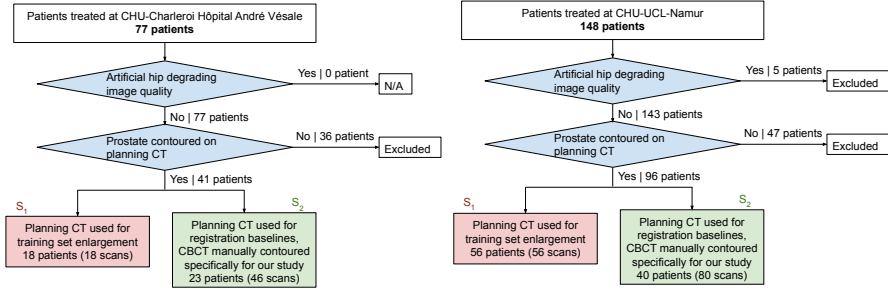


Figure 6.2: Case selection from CHU-Charleroi Hôpital André Vésale and CHU-UCL-Namur.

connected voxels. No disconnected region of the same organ was observed. The main advantage of fully convolutional neural networks is that they output predictions at the same resolution as the input. One output channel was considered per organ. The network was trained with the Dice loss. The Adam optimization algorithm [85] was used with a learning rate of 10^{-4} . The number of epochs was chosen such that convergence was reached. The hyper-parameters mentioned here were the same as in Brion et al. [18] and proved satisfactory on the data used in this work. For this reason and to keep data available for training and testing, no validation set was considered here. Training data were augmented online using rotation (between -5° and 5° along each of the three axes), shift (between -5 and 5 pixels along each axes), and shear (reasonable values for the affine transformation matrix). The batch size was set to two, which was the maximum size affordable on our 11 Gb graphical processing units (GPU).

We performed 3-fold cross-validation with the 63 CBCT scans of set $\mathcal{S}_2$, where 2 folds ($n_{CBCT} \leq 42$ volumes in total) were used as the training set and one fold (21 volumes) as the test set, as shown in Table 6.1. The number of training CBCT scans n_{CBCT} was varied such that $n_{CBCT} \in \{0, 6, 10, 20, 30, 42\}$. The training set was augmented with n_{CT} annotated CT scans from set $\mathcal{S}_1$ such that $n_{CT} \in \{0, 20, 74\}$. The same CT scans were added to the training CBCT scans independently of the considered training folds. Hence, the training set contained $n_{CBCT} + n_{CT}$ volumes in total. Note that the test set contained no CT scans (since our goal was to segment CBCT scans

Table 6.1: Three-fold cross-validation. To train the model, we used n_{CT} CT scans from $\mathcal{S}_1$ and the n_{CBCT} first volumes from the CBCT folds labeled “train.” To test the model, we used all 21 volumes from the CBCT fold labeled “test.”

$\mathcal{S}_1$ (CT)	$\mathcal{S}_2$ (CBCT)		
	fold 1	fold 2	fold 3
train	train	train	<i>test</i>
train	train	<i>test</i>	train
train	<i>test</i>	train	train

only). The source code is publicly available at https://github.com/eliottbrion/pelvis_segmentation.

6.2.3 Validation and Comparison Baselines

In order to evaluate our contouring results, we used four metrics comparing the predicted and manual segmentations. The Dice similarity coefficient (DSC) and the Jaccard index (JI) measure the overlap between two binary masks, while the symmetric mean boundary distance (SMBD) assesses the distance between the contours (i.e., the sets of points located at the boundary of the binary masks) delineating those binary masks. We also computed the difference between the manual and predicted volumes for all the organs considered. More specifically,

$$\text{DSC} = \frac{2|M \cap P|}{|M| + |P|}, \quad (6.1)$$

$$\text{JI} = \frac{|M \cap P|}{|M \cup P|}, \quad (6.2)$$

$$\text{SMBD} = \frac{\overline{D}(M, P) + \overline{D}(P, M)}{2}, \quad (6.3)$$

where M and P are the sets containing the matricial indices of the manual and predicted segmentation 3D binary masks, respectively; $\overline{D}(M, P)$ is the mean of $D(M, P)$ over the voxels of Ω_M ; and $D(M, P) = \{\min_{x \in \Omega_P} \|s \odot (x - y)\|, y \in \Omega_M\}$, where Ω_M , Ω_P are the boundaries extracted from

M and P , respectively, and $s^\top = (1.2, 1.2, 1.5)$ is the pixel spacing in mm. Comparing the manual and predicted organ volumes was motivated by the field of application of this study. Indeed, from the perspective of adaptive radiotherapy, the organs' volumes are needed in order to compare the initial CT plan dose-volume histograms for bladder, rectum, and prostate with the doses actually delivered as determined from CBCT scans acquired during the image-guided treatment [65]. The manual and predicted organ volumes were compared using a Bland–Altman plot, which allows quantification of the agreement between two quantitative measurements (i.e., the manual and predicted organ volumes) by studying their mean difference and constructing limits of agreement [57]. We computed the bias as:

$$\text{Bias} = \frac{1}{n} \sum_{i=1}^n (p_i - m_i), \quad (6.4)$$

where n is the number of patients in the test set and $p_i = s_1 \times s_2 \times s_3 \times |M_i|$, $m_i = s_1 \times s_2 \times s_3 \times |P_i|$ are the volumes of the manual and predicted segmentations of the i^{th} patient. It provides the systematic under- or over-estimation of the predicted volumes. We also computed the precision,

$$\text{Precision} = \frac{1}{n} \sum_{i=1}^n |p_i - m_i|, \quad (6.5)$$

which measures the difference between manual and predicted volume (in absolute value).

The DL-based segmentation was compared with different alternative approaches as summarized in Table 6.3. Two segmentation methods based on deformable image registration (denoted DIR in Table 6.3, second column) were applied to our dataset. First, the contours from the planning CT scans of set $\mathcal{S}_2$ were mapped to the follow-up CBCT scans of the same patient by using a rigid registration followed by DIR with the ANACONDA algorithm without controlling regions of interest (ROIs) in RayStation (<https://www.raysearchlabs.com/raystation/>) (Version 5.99.50.22) [166]. This algorithm adopts an intensity-based registration. Second, the contour was mapped from the planning CT scan to the follow-up CBCT scan using the diffeomorphic morphons' DIR algorithm implemented in OpenReggui (<https://openreggui.org/>) [74].

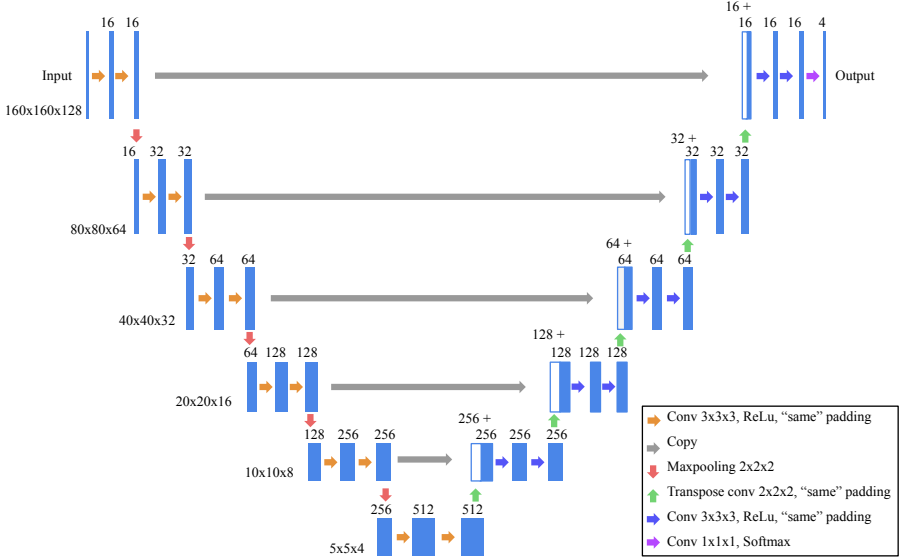


Figure 6.3: 3D U-net model architecture. Each blue rectangle represents the feature maps resulting from a convolution operation, while white rectangles represent copied feature maps. For the convolutions, zero padding was chosen such that the volume size was preserved (“same” padding). The output size was 4: one per segmentation (bladder, rectum, and prostate) and one for the background.

This method exploits the local phase of the image volumes to perform the registration. Therefore, it is suited for registering image volumes with different contrast enhancement, such as CT and CBCT scans. The diffeomorphic version of the algorithm forces anatomically plausible deformations. We also compared our DL method with the Mattes mutual information rigid registration algorithm [116], as implemented in OpenReggui.

6.3 Results

In this section, we assess the performance of our algorithm in terms of overlap (i.e., DSC and JI), distance (i.e., SMBD), and volume agreement measurements. In the first part, we compare the overlaps and

distances measured between our algorithm in different settings and the considered DIR-based segmentation approaches. In the second part, we further evaluate the performance of our best algorithm (i.e., 3D U-net trained with all available CT and CBCT scans) by assessing whether the predicted organ volumes are in good agreement with the volumes determined by manual segmentation. This was done by Bland–Altman analysis.

In Figure 6.4, the DSCs between the segmentation output of the fully convolutional neural network (FCN) and the ground truth segmentation were computed and averaged over all 63 CBCT scans from the three test folds. This was done for different numbers of training CBCT and CT scans. The results were then compared with the RayStation DIR algorithm, the diffeomorphic morphons’ algorithm, and rigid registration. Table 6.3 completes the plots in Figure 6.4 by providing the means and standard deviations of DSC, JI, and SMBD for different numbers of training CBCT scans and different numbers of training CT scans. The statistical model used for comparing the performances was a mixed model with a random intercept on the patient. It showed significant differences between algorithms’ performance for all organs regarding their DSC (bladder, rectum, prostate $p < 10^{-3}$), JI (bladder, rectum, prostate $p < 10^{-3}$), and SMBD (bladder, rectum, prostate $p < 10^{-3}$). In the following paragraphs, the notation $\text{Ours}(n_{CBCT}, n_{CT})$ stands for the 3D U-net proposed in this study with n_{CBCT} CBCT scans and n_{CT} CT scans in the training set. The p -values provided below were obtained by performing a Tukey’s range test on the DSCs. The following observations can be made based on Figure 6.4 and Table 6.3.

First, CBCT scans were more valuable than CT scans to train a CBCT segmentation model. This was not surprising and supported by the observation that a model trained on 40 CBCT and 0 CT scans performed significantly better than a model trained on 0 CBCT and 40 CT scans for all organs (bladder, rectum, prostate $p < 10^{-3}$). The DSCs reached 0.634, 0.286, and 0.525 with $\text{Ours}(0, 40)$ and 0.845, 0.754, and 0.722 with $\text{Ours}(40, 0)$, for bladder, rectum, and prostate, respectively. Furthermore, a model trained only on 74 CT scans reached approximately the same performance as a network trained on only six to 10 CBCT scans for all the organs. Moreover, the more CBCT scans there were in the training set, the higher the DSCs on the test set were.

This result made sense since adding new CBCT scans to the training set allowed the network to generalize on the test set (exclusively composed of CBCT scans) better. More surprisingly, we observed that once a sufficient number (typically 20) of CBCT scans were part of the training set, the benefit of adding CBCT or CT scans was practically the same. Indeed, compared with a model trained on 20 CBCT and 20 CT scans, the model trained on 40 CBCT and 0 CT scans did not lead to a significant improvement in performance (bladder $p = 0.877$, rectum $p = 0.700$, prostate $p = 0.629$). The DSCs reached 0.815, 0.731, and 0.682 with Ours(20, 20) for bladder, rectum, and prostate, respectively. This confirmed that augmenting a CBCT training set with CT scans might be quite valuable.

Second, from the CT perspective, we clearly observed that the more CT scans there were in the training set, the higher the mean DSC became. Indeed, Ours(20, 74) was significantly better than Ours(20, 0) for all organs (bladder, rectum $p < 10^{-3}$, prostate $p < 10^{-2}$). We explained this improvement by the learning of more generic features, leading to better generalization. However, we observed that the difference in the average DSC between Ours(20, 0) and Ours(20, 20) was approximately equal to the difference in average DSC between Ours(20, 20) and Ours(20, 74), whereas 20 new CT scans were added to the training set in the first case and 54 new CT scans in the second case. This may indicate saturation of the performance improvement produced by adding CT scans to the training set. Moreover, when the number of training CBCT scans was large, adding training CT scans improved performance for rectum only ($p < 0.01$): no statistically significant incremental change in performance was observed for bladder or prostate ($p = 0.780$ and $p = 0.630$, respectively) when Ours(42, 74) and Ours(42, 0) were compared. A plausible interpretation was that most of the useful information present in the CT scans was already captured in the relatively large CBCT training set. More importantly, in line with our objective of limiting the annotation of CBCT scans, we observed that the performance obtained with 42 CBCT and 0 CT scans could be reached with 20 CBCT and 74 CT scans for all organs (bladder $p = 0.940$, rectum $p = 0.882$, prostate $p = 0.994$). Hence, the availability of 74 annotated CT scans reduced the number of annotated CBCT scans significantly (by a factor of approximately two).

Third, when all available CT and CBCT scans (42 CBCT and 74 CT scans) were used for training, our approach significantly outperformed the rigid registration, RayStation DIR algorithm, and diffeomorphic morphons' algorithm for bladder and rectum ($p < 10^{-3}$), but not for prostate ($p = 0.911$). These conclusions are illustrated on a representative patient in Figure 6.5. The results also showed that the rigid registration was outperformed by the ANACONDA algorithm, which was in turn outperformed by the diffeomorphic morphons' algorithm for bladder and rectum. As mentioned above, both DIR methods were statistically similar to the rigid registration approach when it came to segmenting prostate. This supported the hypothesis that prostate underwent less deformation than bladder and rectum, which were subject to regular influxes and voiding of matter. Although our analysis was based on the DSC, both JI and SMBD led to the same conclusions.

Figure 6.6 presents Bland–Altman plots comparing the organ volumes reached manually and by our DL-based predictions (obtained with Ours(42, 74)), using the bias, precision, and 95% limits of agreements (LoA). The bias normalized by the manual volume was below 5 % for all organs (bladder 4.78%, rectum 1.21%, prostate 2.51%). The precision normalized by the manual volume was similar for bladder and rectum (bladder 13.3%, rectum 13.9%) and larger for prostate (27.9%). The LoA of bladder were also close to the LoA of rectum (−32% and 41% for bladder and −33% and 35% for rectum), whereas they were larger for prostate (−65% and 70%). Table 6.2 completes the Bland–Altman plots by providing the means and standard deviations for the manual and predicted organ volumes. Moreover, a one-sample t -test was performed on the differences between the manual and predicted volumes normalized by the manual volume for each organ. The resulting p -values for all organs are presented in Table 6.2 and were not significantly different (bladder $p = 0.285$, rectum $p = 0.897$, prostate $p = 0.438$). This meant that the predicted and manual contours were similar in means according to the t -test.

Computational cost analysis was performed by measuring the running time on our machine equipped with an 11Gb GeForce GTX 1080 Ti graphics card. The rigid registration of one image ran in 1.05 min. The deformable image registration with the ANACONDA and morphons' algorithms ran in 1.92 min and 8.33 min, respectively. The in-

Table 6.2: Absolute and relative differences between manual and predicted organs volumes. p -values are calculated using a one-sample t -test on percentage differences.

Organ	Volumes ($\times 10^4$ mL)		Differences between Manual and Predicted Volumes				
			Absolute ($\times 10^4$ mL)		Percentage (%)		
	Manual	Predicted	Bias	Precision	Bias	Precision	p -Value
Bladder	21.9 $\pm$ 12.9	20.7 $\pm$ 11.4	1.18	2.46	4.78	13.3	0.285
Rectum	5.96 $\pm$ 1.66	5.87 $\pm$ 1.55	0.094	0.826	1.21	13.9	0.897
Prostate	5.87 $\pm$ 2.98	5.53 $\pm$ 2.07	0.340	1.64	2.51	27.9	0.438

ference time for one image with the DL approaches was much lower. It reached 0.15 s independently of the learning strategy. Indeed, the number of images in the training set had no impact on the inference time. The training time needed to reach convergence depended on the size of the training set. Hence, Ours(20, 0), Ours(20, 74), Ours(42, 0), and Ours(42, 74) were trained in 17.3, 224, 167, and 220 min, respectively.

6.4 Discussion

Based on Table 6.3 (first part) and Figure 6.4, the 3D U-net approach was the most satisfactory approach for automatic segmentation of bladder and rectum on CBCT scans. This supported the initial hypothesis that registration-based approaches failed in the case of large deformation and alternative approaches using the statistics of the target image (i.e., the CBCT scan) were more suitable. This observation was also consistent with the state-of-the-art algorithms shown in Table 6.3 (second part), where DL approaches outperformed alternative approaches for bladder and rectum.

Still based on Table 6.3 (first part) and Figure 6.4, the 3D U-net slightly outperformed the registration-based approaches for prostate, but this improvement was not statistically significant. 3D U-net’s lower performances for prostate than for bladder and rectum was further supported by the Bland–Altman analysis of the manual and predicted volumes. Indeed, this analysis provided less than 5% bias for all organs, but higher precision (i.e., a larger spread of the predictions, as defined in (6.5)) for prostate than for bladder and rectum. Furthermore, most

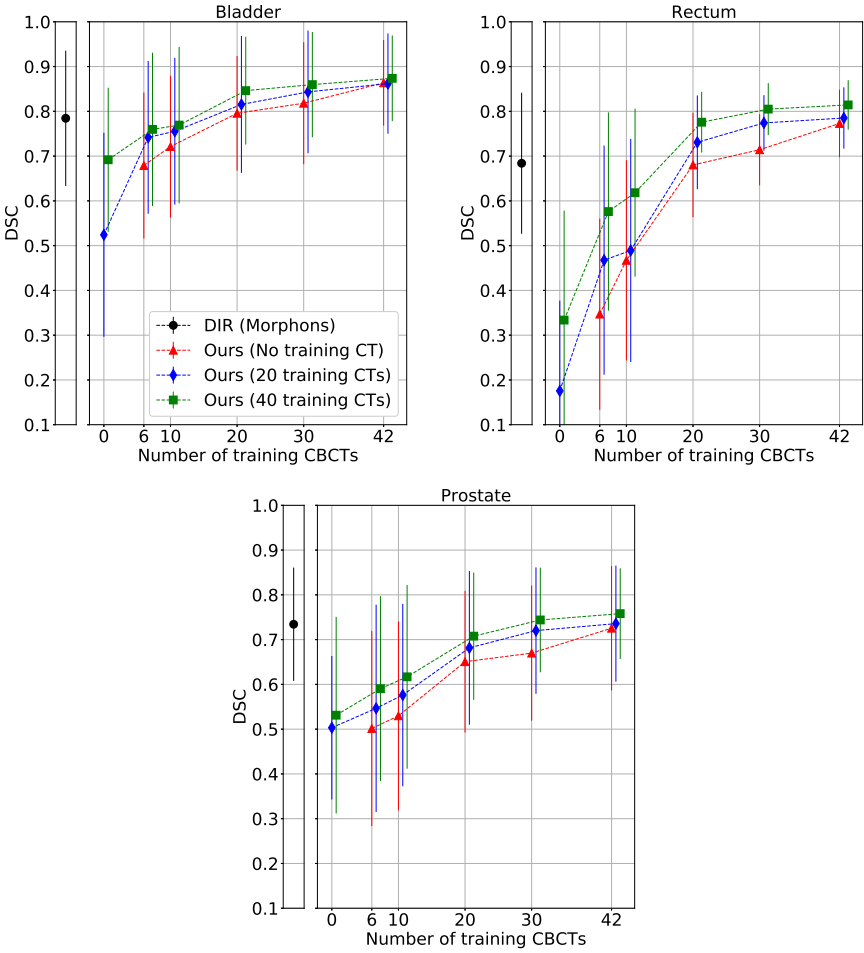


Figure 6.4: Influence of the number of training CBCT and CT scans on the DSC. Bars indicate one standard deviation for the group of 63 patients. DSC: Dice similarity coefficient.

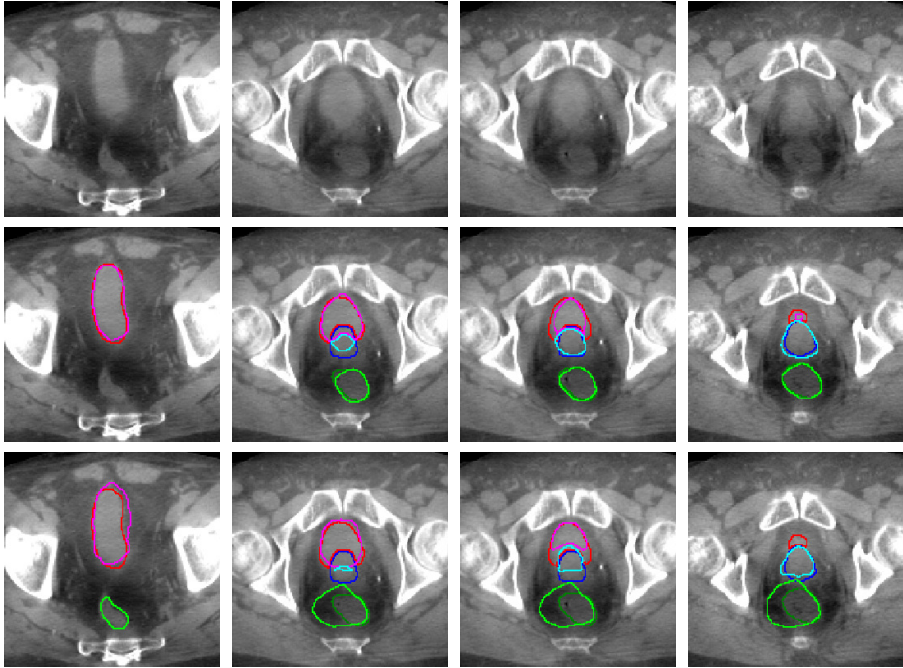


Figure 6.5: Comparison of manual, 3D U-net, and morphons' DIR-based segmentation for a representative patient. Each column corresponds to a slice of the same CBCT scan. Dark colors represent reference segmentations (both second and third rows), while light colors show 3D U-net segmentation (second row) and morphons' DIR-based segmentation (third row). The predicted bladder, in pink, has a DSC of 0.940 (U-net) and 0.864 (morphons); rectum, in light green, has a DSC of 0.791 and 0.759; the prostate, in light blue, has a DSC of 0.780 and 0.730.

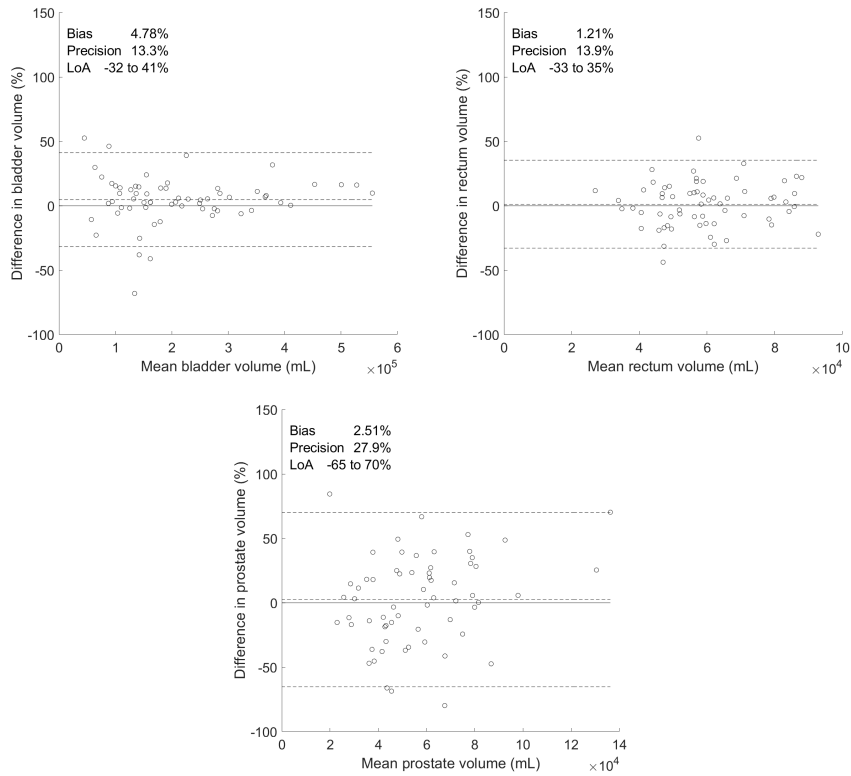


Figure 6.6: Bland–Altman plots for bladder, rectum, and prostate derived from the differences between the predicted and manual segmentations. The solid lines represent no difference; the dotted lines depict the mean difference (bias) and 95% limits of agreements (LoA).

Table 6.3: DSC, JI, and SMBD, between the manual contours and the output of our proposed algorithm in different settings (number of training CBCT scans, number of training CT scans) for bladder, the rectum, and prostate. Comparison with other benchmarking algorithms. DL: deep-learning, RS: RayStation, DSC: Dice similarity coefficient, JI: Jaccard index, SMBD: symmetric mean boundary distance, DIR: deformable image registration, PSM: patient specific model. * Evaluated on a dataset different from ours. † Results reported on a test set containing both CBCT and CT scans. ‡ The authors computed the root mean squared boundary distance rather than the SMBD. § The authors computed the mean boundary distance and not the SMBD.

Study	Method	DSC			JI			SMBD (mm)		
		Bladder	Rectum	Prostate	Bladder	Rectum	Prostate	Bladder	Rectum	Prostate
Ours	DL ($n_{CBCT} = 20, n_{CT} = 0$)	0.796 ± 0.128	0.680 ± 0.117	0.651 ± 0.158	0.677 ± 0.153	0.526 ± 0.123	0.501 ± 0.164	3.94 ± 2.18	3.85 ± 1.39	4.90 ± 2.85
Ours	DL ($n_{CBCT} = 20, n_{CT} = 74$)	0.846 ± 0.120	0.776 ± 0.068	0.708 ± 0.142	0.749 ± 0.155	0.638 ± 0.086	0.565 ± 0.157	3.02 ± 2.26	3.14 ± 1.43	3.87 ± 2.19
Ours	DL ($n_{CBCT} = 42, n_{CT} = 0$)	0.864 ± 0.096	0.773 ± 0.075	0.725 ± 0.139	0.771 ± 0.131	0.636 ± 0.098	0.585 ± 0.151	2.77 ± 1.95	3.06 ± 1.55	3.51 ± 2.03
Ours	DL ($n_{CBCT} = 42, n_{CT} = 74$)	0.874 ± 0.096	0.814 ± 0.055	0.758 ± 0.101	0.787 ± 0.131	0.690 ± 0.077	0.620 ± 0.120	2.47 ± 1.93	2.38 ± 0.98	3.08 ± 1.48
DIR	Rigid image registration	0.714 ± 0.149	0.646 ± 0.090	0.730 ± 0.108	0.576 ± 0.175	0.484 ± 0.102	0.585 ± 0.124	6.93 ± 4.09	5.30 ± 1.91	3.81 ± 1.44
DIR	DIR, RS intensity-based	0.737 ± 0.155	0.662 ± 0.100	0.739 ± 0.110	0.606 ± 0.187	0.504 ± 0.115	0.597 ± 0.127	6.27 ± 4.08	5.08 ± 2.04	3.61 ± 1.42
DIR	DIR, morphons	0.784 ± 0.151	0.684 ± 0.158	0.734 ± 0.127	0.668 ± 0.182	0.539 ± 0.165	0.594 ± 0.143	5.04 ± 3.90	5.00 ± 3.43	3.65 ± 1.64
Schreier et al. (2019) [145] *	DL ($n_{CBCT} = 300, n_{CT} = 300$)	0.932 †	0.871 †	0.840 †	-	-	-	2.57 ± 0.54 †,‡	2.47 ± 0.64 †,‡	2.34 ± 0.68 †,‡
Brion et al. (2019) [18] *	DL ($n_{CBCT} = 32, n_{CT} = 64$)	0.848 ± 0.085	-	-	0.745 ± 0.114	-	-	2.8 ± 1.4	-	-
Hänsch et al. (2018) [62] *	DL ($n_{CBCT} = 124, n_{CT} = 88$)	0.88	0.71	-	-	-	-	-	-	-
Motegi et al. (2019) [123] *	DIR, MIM intensity-based	~ 0.80	~ 0.40	~ 0.55	-	-	-	-	-	-
Motegi et al. (2019) [123] *	DIR, RS intensity-based	~ 0.78	~ 0.70	~ 0.75	-	-	-	-	-	-
Takayama et al. (2017) [153] *	DIR, RS intensity-based	0.69 ± 0.07	0.75 ± 0.05	0.84 ± 0.05	-	-	-	-	-	-
Woerner et al. (2017) [168] *	DIR, cascade MI-based	~ 0.83	~ 0.77	~ 0.80	-	-	-	-	-	-
König et al. (2016) [90] *	DIR, rigid on bone and prostate	0.85 ± 0.05	-	0.82 ± 0.04	-	-	-	~ 2.6 §	~ 2.3 §	~ 2.3 §
Thor et al. (2011) [156] *	DIR, demons	0.73	0.77	0.80	-	-	-	-	-	-
van de Schoot et al. (2014) [162] *	Patient specific model	~ 0.87	-	-	-	-	-	-	-	-
Chai et al. (2012) [24] *	Patient specific model	0.78	-	-	-	-	-	-	-	-

other state-of-the-art DIR-based algorithms outperformed our approach for prostate. This showed that DIR-based approaches were still valuable in situations with limited organ deformation and where poor contrast made the use of vanilla DL models challenging. A first way to improve the segmentation results for prostate and outperform DIR-based approaches without annotating more CBCT scans might be to generate pseudo CBCT scans as in Schreier et al., but our study showed that increasing the number of already annotated CT scans further was a valuable alternative, albeit with a risk of saturation. If few data are available, a second option could be to promote a desired shape or structure in the deep model prediction [130, 141]. A third option could be to perform unsupervised domain adaptation [80]. This approach requires annotations in a source domain (CT), but not in the target domain (CBCT). This will be the subject of future research.

From an application point of view, the study showed that the more CBCT scans were contoured, the better the DSC on the predicted contours. However, contouring CBCT scans is not part of the clinical workflow, is time consuming, and is not easy because of the poor contrast between the different regions of interest. Hence, we showed that expanding the training set with CT scans improved the segmentation performances for all considered organs, especially when few contoured CBCT scans were available. Our 3D U-net that reached the best segmentation performances was trained with 42 CBCT and 74 CT scans.

Most cases of failure were observed for prostate, which had the lowest DSC of the organs. This may be due to the fact that prostate is hard to see on CBCT scans and often pushes on bladder, as we can see in Figure 6.5. Hence, some upper parts of prostate were often wrongly classified as bladder, which decreased the DSC for prostate. Since bladder is larger than prostate, misclassification at the boundary between the two organs had less impact on the DSC of bladder. A second case of failure occurred at the top and bottom slices of rectum, which was wrongly classified as the background (or inversely, the background was wrongly classified as rectum). This made sense since there were few differences in contrast between rectum, anal canal, and colon. The impact of such errors on prostate and rectum, as well as the required contour quality for clinical use in adaptive radiotherapy, was such that additional quality assessment with a contours review process was needed. This should be

done by radiation oncologists and will be the subject of future research.

Our DL approach also outperformed or achieved the same performance as patient specific models for bladder. Those models rely on PCA to extract principal modes of deformation from landmarks placed on bladder's contour and across several contoured images for each patient being considered. The drawback for clinical use of such approaches is that (i) a different model is required for every patient and organ and (ii) several images per patient are needed to build the model.

Concerning alternative DL methods, the current work slightly outperformed our initial conference paper, Brion et al. [18], on bladder segmentation with 3D U-net. This was probably due to the larger training database and/or the multi-class formulation used in this work, since three organs were segmented instead of one. Only 41 of the patients used in our conference paper were kept in this study. This was because the remaining patients had either had their prostates removed or lacked fully annotated scans. New patients were also added. The two datasets were thus different. However, Schreier et al.'s work was the closest to this study. Hence, we did a more thorough comparison with their findings. They obtained a higher DSC than we did for all the organs considered in this study. This might be explained by the fact that they used more samples in their training set (300 CT and 300 pseudo CBCT scans compared with 74 CT and 42 CBCT scans). However, it was hard to determine whether this was the only explanation for their better results. Indeed, in Figure 6.4, we see that the DSC increased more slowly as the number of training samples increased. Interestingly, they ran the patch-wise 3D U-net proposed by Hänsch et al. on their test set and obtained DSCs of 0.927, 0.860, and 0.816 for bladder, rectum, and prostate, respectively. Those results were higher than the results obtained on bladder ($\text{DSC} = 0.88$) and rectum ($\text{DSC} = 0.71$) by Hänsch et al. Therefore, their test set might be of a higher quality than ours, which could be a limitation of their approach in clinical practice, where low quality images are common. Another shortcoming is that they reported their results on a dataset that included both CBCT and CT scans (10%). It was therefore unclear how well their method would perform on a dataset containing only CBCT scans (such as ours). As a final remark, their proposed generation of pseudo CBCT scans from clinically contoured CT scans was a powerful tool for solving the problem of CBCT annotations. However,

such knowledge of artificial data generation might not be present in all hospitals. To summarize this comparison, we considered the two publications to be complementary, with our strengths being the size of our test set, detailed comparison with registration approaches, and the detailed study of the impact of additional CT scans in the training database.

6.5 Conclusions

In this work, a 3D U-net DL model was trained on CBCT and CT scans in order to segment bladder, rectum, and prostate on CBCT scans. The proposed approach significantly outperformed all the DIR-based segmentation methods applied on our dataset in terms of DSC, JI, and SMBD for bladder and rectum. The conclusions were more mitigated concerning prostate, where the DL-based segmentation did not significantly outperform alternative approaches. A Bland–Altman analysis on the manual and predicted organs’ volumes revealed a low bias on the predicted volumes for all organs, but higher precision (i.e., a larger spread of the volumes) for prostate than for the other organs. Furthermore, the study showed that the cross-domain data augmentation consisting of adding CT to the CBCT scans in the training set significantly improved the segmentation results. A further step will be to highlight these improvements by showing the better tumor coverage and reduction in the doses delivered to organs at risk that it allows.

Chapter 7

Domain adversarial networks and intensity-based data augmentation for male pelvic organ segmentation in Cone Beam CT

In Chapters 5 and 6 we proposed a method sharing knowledge between CT and CBCT to automatically contour the bladder, rectum, and prostate on CBCT. A limitation of this method, called cross-domain data augmentation, was that a few annotated CBCTs are required in the training set. In this chapter, we propose to use other knowledge propagation-based methods for contouring these three organs on CBCT requiring annotations only on CT: one based on adversarial networks, and the other on intensity-based data augmentation.

In the article, we claim that the approach based on virtual CBCT generation has the drawback of requiring thorough expertise in the field. An even stronger constraint that is not mentioned is that the physical model may require difficult calibration (e.g., determine the spectrum of the X-ray tube). This point further motivates our approach not to rely

on a physical model.

An additional baseline could be to train a neural network with both annotated CTs and annotated CBCTs. This corresponds to the cross-domain data augmentation of the previous chapter. Applying this method to the dataset of this chapter (for a fair comparison) is left as future work.

In this chapter, domain adversarial networks perform less well than morphons. One possible reason for the latter’s superiority is that the registration algorithm disposes of a powerful prior: the anatomy at the moment of planning. This suggests a way to improve neural networks: by including the a priori of the planning CT anatomy. This is linked to Chapter 4, but in this case, the propagation would not be spatial (from one slice to another) but temporal (from one instant to another). Similar to what has been performed in Chapter 4, the a priori could be encoded as an additional input channel. However, experiments not described in this thesis have shown that including the volumetric planning CT contours as a prior in u-net did not improve performance compared with an “image only” input. Figuring out why this is the case is left for future work.

The domain adversarial approach investigated in this chapter looks at features at a given level in the segmentation network and tries to make them domain-agnostic. In this chapter, we compare the impact of the level at which these features are chosen on segmentation performance. In other experiments not described in this chapter, we also tried to adapt multiple levels at the same time. However, this was very computationally intensive and preliminary investigations on a limited number of hyperparameters did not show significant improvement.

In this thesis, data augmentation has been performed in three different ways: (i) geometric (translation, rotation, scaling), (ii) cross-domain, and (iii) intensity-based. Another kind of data augmentation motivated by the application could be to perform morphon registrations to images in the training set to generate additional plausible shapes. This can be considered an instance of geometric data augmentation and is also left for future work.

7.1 Introduction

About half of cancer patients are currently treated with radiotherapy. This treatment consists in delivering a prescribed dose of radiation, typically X-rays, to the tumor, while limiting the dose delivered to the surrounding healthy organs, also called organs at risk or OARs. The spatial distribution of the dose is optimized at treatment planning, which requires a computed tomography (CT) scan of the patient and the delineation of the tumor and OAR contours on the scan in order to evaluate the dose delivered to those regions. In current practice, trained clinicians delineate the contours manually on the CT. Then, the computed dose is split into different fractions, which are delivered to the patient during successive treatment sessions. However, variations in patient anatomy between sessions can impair dose conformity, with typically too large a dose delivered to the healthy organs or too low a dose to the tumor [134]. Those daily variation are particularly large in the pelvic region due to physiological function (e.g., bladder and rectal filling or voiding). This can jeopardize treatment optimality, with possibly lower tumor control probability or more acute secondary effects, affecting quality of life, such as anal canal bleeding for prostate cancer.

Adaptive radiation therapy is aimed at ensuring an optimal treatment by using daily images to monitor anatomical modifications and adapt the treatment plan accordingly. For this purpose, drawing new contours on this daily image is a crucial step [56, 138]. Since drawing those contours manually takes too long to be carried out before each treatment session (2-4 hours per volumetric image, with one image for each of the ~ 20 treatment sessions), an automated segmentation algorithm is needed. Registration-based segmentation approaches are widely used for medical image segmentation due to their easy interpretation and high performance in regions with little anatomical variations [129, 142]. However, these methods fail in the case of large deformations between scans to be registered, as it is often the case in the pelvic region [156, 174]. Furthermore, they are rather slow, reaching 1.92 min per image for the ANACONDA algorithm [166] and 8.33 min per image for the diffeomorphic morphons algorithm, which is suited for registering planning and daily images [74].

While deep neural networks such as U-Net provide state-of-the-art performance for medical image segmentation, their use in our application is not direct. Indeed, it consists of a supervised learning model whose internal parameters must be adjusted on a set of already contoured images. Although such databases are available for CT scans, since contouring them has been part of classical clinical routine, the use of helical CT in the treatment room to position the patient is challenging because physical constraints prevent the CT scanner and the dose delivery machine from being in the same place. Therefore, the daily image is often what can be seen as a noisy CT scan: a cone beam CT (CBCT) scan. Since these images are not contoured in clinical routine, there are no (or few) labeled CBCT scans on which a deep learning model can be trained for CBCT segmentation. This raises the question of how to leverage the large database of annotated CT scans in order to build a model able to segment images from a slightly different domain: CBCT scans (see Fig. 7.1).

The most direct way to do this is simply to train a network on a mixed training set made of both labeled CT scans and a small number of CBCT scans contoured specifically for model training [18, 62, 101]. In order to avoid the annotations of CBCT scans, pseudo-CBCT scans can be generated from annotated CT scans [145] and added to the CT scan training set. However, those approaches need either a small number of annotated CBCT scans or a physical model for the pseudo-CBCT scan generation, which requires thorough expertise in the field. In this paper, we report on training a deep neural network for male pelvic organs segmentation in CBCT scans using labeled CT and unlabeled CBCT scans, as available in the current treatment workflow, and limited physical expertise (i.e., it does not require deriving an explicit physical model for CT to CBCT conversion). For this purpose, we propose two different strategies. On the one hand, we follow an unsupervised domain adaptation (UDA) learning strategy where the CT and CBCT scans are considered the source and target domains, respectively. In machine learning, unsupervised domain adaptation refers to strategies that train a model using labeled examples, a "source domain" (in our case, CTs), and only unlabeled examples in a similar yet different "target domain"

(in our case, CBCTs). We extend the method proposed in [53, 80] for learning domain-invariant features to a U-Net architecture. This lets us train a segmenter using the annotated CT scans while forcing the features to be transferable to the CBCT scans. On the other hand, we use standard (non adversarial) training with intensity-based data augmentation on the CBCT inputs in order to bring CT and CBCT intensity distributions closer at the input level and enhance the learning of domain-invariant features. Then, the selected best method for intensity-based data augmentation is combined with adversarial training. In addition, the effect of database size on the model’s performance is analyzed for the intensity-based data augmentation approach. Our contribution is twofold:

- Adversarial networks for unsupervised domain adaptation are implemented with 3D U-Net and applied to male pelvic organ segmentation on CBCT, including the bladder, rectum, and prostate. The model is trained using annotated CTs and non-annotated CBCTs, outperforming (for each of the three organs) a model trained only on CT (i.e., without adversarial networks). Our architecture for unsupervised domain adaptation with a gradient reversal layer uses (i) a 3D end-to-end architecture (in contrast to slice-wise 2D ones) and (ii) U-Net as its base (while some previous works use simple autoencoders or patch-based methods, without skip connections). Our code is made public and can be used for any segmentation application experiencing a shift of distribution between training and test images, a recurring problem in biomedical imaging.
- Different strategies for intensity-based data augmentation are compared and we demonstrate that brightness, linear, and bilinear transformations improve generalization of CT-only models to CBCT data without requiring annotated CBCTs. For these strategies, neural networks are trained in a standard, non adversarial fashion. Applying intensity-based data augmentation during training achieves similar results to deformable image registration (DIR) for the prostate and outperforms existing automatic segmentation methods (trained with labeled CTs and unlabeled CBCTs) for the bladder and rectum. Moreover, our model is several orders of mag-

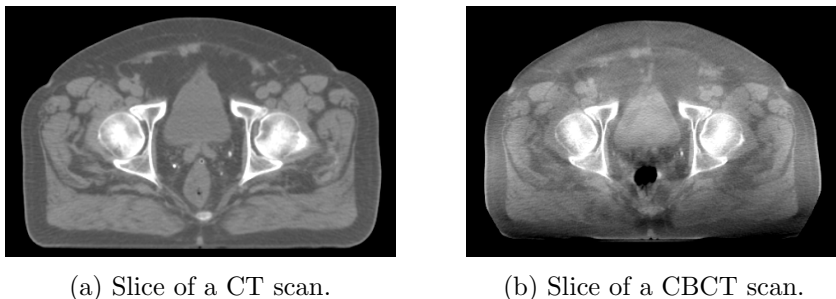


Figure 7.1: Radiotherapy relies on CT scans for treatment planning and on cone beam CT (CBCT) for treatment monitoring (these can be seen as noisy CTs). Current clinical workflow generates large databases of annotated CT scans on which neural networks can be trained for segmentation tasks. However, using these segmentation networks on CBCT scans at test time requires innovative methods due to differences in appearance between the two modalities.

nitude faster than DIR at inference, which is essential for online treatment adaptation. It also outperform a cycleGAN-based approach.

7.2 Related works

7.2.1 Deep domain adaptation

While medical imaging datasets are becoming more available, the lack of annotated data (i.e., labels) remains a challenge for supervised machine learning algorithms. Therefore, different approaches have been proposed in order to learn with less supervision. Among those methods, transfer learning considers learning from related learning problems. In particular, domain adaptation is a transfer learning category where the training and test data are from different domains whereas the task performed by the model remains similar for both domains [27]. The source domain refers to the domain where labels are available and supervised learning can be performed. The target domain refers to the domain where no or few labels are available and where the actual task should be performed.

When few labels are available in the target domain, supervised learning can be used in both source and target domains. Domain adaptation with fine-tuning strategies has been studied and applied experimentally for the segmentation of brain lesions [55]. Also, cross-domain data augmentation has been proposed in order to segment male pelvic organs in CBCTs. This approach uses labeled CTs to augment the training set [101]. However, such approaches still require some labels in the target domain, which limits their use in applications where target labels are hard to collect. If no labels are available in the target domain, unsupervised domain adaptation (UDA) is considered. We generally divide current UDA methods into three categories: feature-level transferring, image-level transfer, and label-level transfer.

7.2.2 Feature-level transferring

Feature-level transferring was adopted by the first deep domain adaptation approaches and attempts to force the extraction of domain-invariant features. Those methods extend deep convolutional networks to domain adaptation either (i) by minimizing the maximum mean discrepancy (MMD) between the features learned from the source and target domains with one or more adaptation layers [110,161], (ii) by adding a domain discriminator determining whether the learned deep features are extracted from source or target data following a domain-adversarial learning strategy [7, 53, 160], or (iii) by minimizing the correlation distance between features [152]. Building domain-invariant features can also rely on disentanglement strategies to build a shared domain-invariant content space and a domain-specific style space [172]. However, feature-level transferring techniques suffer from several limitations, especially when it comes to a segmentation task. First, the semantic consistency is not enforced through the alignment of the marginal distributions [177]. Second, the level of a deep representation to be aligned is not trivial to choose and not easily interpretable, which can lead to a tedious tuning process. Indeed, different choices of feature levels may result in different segmentation results. To tackle those limitations, a significance-aware feature purification before the adversarial adaptation was proposed recently [114], which eases the feature alignment and stabilizes the adversarial training course. Medical image segmentation has been performed using adver-

sarial UDA between MR images [80] and between MR images and CT scans [42]. Adversarial UDA has also been used to segment MR images with both semantic and boundary information [104].

7.2.3 Image-level transferring

Image-level transferring consists in adapting the source images to look as if they belong to the target domain [14]. Hence, the distribution alignment is performed in the raw pixel space. Domain adaptation can be formulated as the generalization to a new domain. From this model generalization perspective, data augmentation techniques can be used to make the source training samples look closer to the target samples. Image data augmentation for deep learning has been reviewed in [148] and, with a focus on medical image segmentation in [125]. Common data augmentation techniques consider affine, elastic and pixel-level transformations. Pixel-level transformation are especially useful for medical image processing since they can simulate the acquisition with different devices. Common transformation are noise injections, modification of the brightness, sharpening or blurring with a kernel and mixup [115, 119, 126, 163]. Such methods are easy to implement and fast but may not be able to produce sufficiently realistic images in order to allow domain adaptation.

In order to cope with those limitations, approaches allowing artificial images generation have been proposed. Image-to-image translation techniques, such as CycleGAN [181], can generate visually appealing results that maintain local content in natural scenes. Image-to-image transferring has been used prior to segment abdominal organs [71], lung tumors [78], cardiac [26, 178] and knee [108] structures using domain adaptation between MR images and CT scans. It has also been used for chest X-ray [25] and fundus images [179] segmentation. Besides, a cycle-and shape-consistent GAN has been proposed for segmentation of cardiovascular and abdominal structures in CT scans and MR images, as well as mammography X-rays [21]. However, image-level transferring techniques also have several limitations. Indeed, for image alignment, the quality of the synthetic images is not guaranteed without large amounts of data from the target domain. Furthermore, those approaches tend to correctly deal with pixel or low-level domain shifts but sometimes fail to capture high-level semantic information [68]. Some

researchers attempt to overcome the limitations of feature and image alignment strategies by combining them for natural [68] and medical images [26]. In [26], cardiac substructures are segmented in MR images and CT scans. Alternatively, approaches attempt to improve the shape and appearance of the structures of interest [9,33,78,178]. Recently, [77] models the co-dependency between images and their segmentation in order to improve both geometry and appearance consistency through the domain transfer process.

7.2.4 Label-level transferring

Label-level transferring consists in predicting pseudo-labels on target data and using them to generalize to the target domain. Self-training considers a model trained on the labeled source data to predict labels on the target data, then simply trains on some of those estimated pseudo-labels (e.g., the most confident ones) [83,106,183]. Self-ensembling uses a teacher and a student network, where the teacher model is an ensemble model obtained by averaging the weights of the student model over the training phase. More precisely, the student network is trained such that it produces predictions that are consistent with the teacher predictions on the target data. In this framework, the predictions of the teacher model are considered as pseudo-labels as the teacher model is an ensemble model, supposed to show good generalization capabilities. Even if current self-ensembling techniques are effective in a semi-supervised setting [154], these approaches need much manually-tuned data augmentation in order to succeed in domain alignment tasks [51]. Concerning semantic segmentation, appropriate data augmentation should be performed to avoid a spatial misalignment between the student and teacher predictions and is a limitation of the approach. Medical image segmentation in MR images based on self-ensembling was first performed by [135], followed by [127]. In [28,83], data augmentation is based on a GAN and allows improved semantic segmentation with self-ensembling. Besides self-training and self-ensembling, multi-view co-training has also been used for medical image segmentation and domain adaptation [169].

7.2.5 CBCT segmentation

Image registration allows contours to be propagated from planning CT to CBCT scans [16,123,153,156,174]. However, those methods fail when large anatomical deformations occur between CT and CBCT scans. Therefore, more sophisticated registration-based approaches have been proposed. This includes deformable image registration (DIR) based on B-splines and mutual information [168]. This last approach uses six successive DIRs with different resolutions and, after visual examination, an optional last pass focusing around the region of interest. Another approach implements a DIR strategy where the deformation is rigidly constrained for bone and/or the prostate, while surrounding tissue can still deform elastically [90].

To eliminate registration-related errors, statistical shape models have been proposed for organ segmentation on CBCT scans [24,162]. However, building such patient-specific shape models require several delineated CBCT scans and landmarks or meshes. More recently, deep learning approaches have been used for the segmentation of CT scans [23,82]. The 3D U-Net fully convolutional neural network has been used with CT and CBCT scans in the training set to segment female [62,64] and male pelvic organs on CBCT scans [18,101]. In order to avoid annotating CBCT scans, pseudo-CBCT scans have been generated from annotated CT scans in order to build a training set including both CT and pseudo-CT scans [145]. In order to improve the segmentation of soft tissues, [103] proposed a method to generate synthetic MR images from CBCT scans using a Cycle-GAN network [180]. Then the synthetic MR images are segmented using manual annotations obtained on the original CBCT scans as ground truth. This is extended in [52], where both CBCT scans and CBCT-generated synthetic MR images have been used to segment male pelvic organs and femoral heads.

7.3 Materials and methods

In this section, we introduce the dataset and the two main strategies investigated in this study: adversarial networks and intensity-based data augmentation. We also present the metrics used to assess the perfor-

mance of the different methods, as well as the deep learning and registration baselines.

7.3.1 Data and preprocessing

In this study, we use data from 134 different patients who underwent radiotherapy for prostate cancer in two different hospitals (41 patients at CHU-Charleroi Hopital André Vésale and 93 patients at CHU-UCLouvain-Namur). For the purposes of the study, we shuffled patients from both hospitals and randomly split them in five cohorts, including four cohorts of 30 patients, denoted $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$ and one cohort of 14 patients, denoted $\mathcal{C}_5$. We collected CT scans for cohorts $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_4$, and $\mathcal{C}_5$ for a total of 104 CT scans. We collected the CBCT scans for cohorts $\mathcal{C}_3$ and $\mathcal{C}_4$ for a total of 60 CBCT scans. The CBCT scans coming from CHU-Charleroi Hopital André Vésale were acquired with a Varian TrueBeam STx version 1.5, while those coming from CHU-UCLouvain-Namur have been acquired with a Varian OBI Cone Beam CT. The use of these retrospective, anonymized data for this study has been approved by both hospitals' ethics committees (May 24, 2017, for CHU-Charleroi Hopital André Vésale and May 12, 2017, for CHU-UCLouvain-Namur). In order to ensure data uniformity across the entire dataset, all the 3D CT and CBCT scans (as well as the 3D binary masks representing the manual segmentations) were re-sampled on a $1.2 \times 1.2 \times 1.5$ mm regular grid. All re-sampled image volumes and binary mask volumes were cropped to volumes of $160 \times 160 \times 128$ voxels containing the bladder, rectum, and prostate.

The only processing applied to the scans prior to deep learning was normalization by the mean and standard deviation. More precisely, we computed the mean μ_{train} and the standard deviation σ_{train} over all pixels of all CT and CBCT scans used in the training set, indifferently. Before both training and inference, we subtracted from each CT or CBCT scan the mean μ_{train} and then divided by the standard deviation σ_{train} .

7.3.2 Adversarial networks

The feature-level UDA follows the adversarial learning strategy proposed in [80] and was implemented using the gradient reversal layer (GRL) ap-

proach proposed in [53]. A GRL is a custom layer for which the gradient is hard-coded to achieve a desired weights update rule (i.e., different from the weight update rule of "standard" backpropagation). A segmenter is trained in the source domain and a domain classifier connected to the intermediate features of the segmenter encourages those features to be domain-invariant. Hence, the segmenter is further split into a feature extractor, which computes the features Lz provided as input to the domain classifier, and a label predictor, which exploits these features, in addition to a few upper skip connections – except for $z = 1$ (first layer) and $z = 11$ (last layer) – to produce a segmentation mask. The feature extractor is denoted $G_f(\cdot; \theta_f)$, the label predictor, $G_y(\cdot; \theta_y)$, and the domain classifier, $G_d(\cdot; \theta_d)$, where θ_f , θ_y , and θ_d are the weights of the feature extractor, label predictor, and domain classifier, respectively. The segmentation is performed by the composition of the feature extraction and the label prediction. Formally, we define the segmenter as $G_{\text{seg}}(\cdot; \theta_f, \theta_y) = G_y(G_f(\cdot; \theta_f); \theta_y)$. The segmenter is implemented by following a classical U-Net architecture. The proposed architecture is shown in Fig. 7.2. The features computed at the output of every layer of U-Net are denoted by $L1, L2, \dots, L11$. The hyperparameter z defines the layer Lz at which the label predictor and domain classifier are connected to the feature extractor.

Both the segmenter and domain classifier are updated with every batch of data. The first half of the batch contains annotated CT scans from patient cohort $\mathcal{C}_1$ (i.e., source data), whereas the second half of the batch contains unlabeled CBCT scans for patient cohort $\mathcal{C}_3$ (i.e., target data). Note that they are unpaired data (i.e., each batch contains at most one image of any given patient). The annotated CT scans are used to update the weights of the feature extractor and label predictor following a classical end-to-end supersized segmentation setting. Additionally, the CT and CBCT scans are used to train the feature extractor and domain classifier following an adversarial strategy. Following [53] and [80], this is achieved by finding the saddle point of the following function:

$$L_{\text{tot}}(\theta_f, \theta_y, \theta_d) = \sum_{\substack{i=1..N \\ d^i=0}} L_{\text{seg}}(\hat{\mathbf{Y}}^i(\theta_f, \theta_y), \mathbf{Y}^i) + \sum_{i=1..2N} L_{\text{dom}}(\hat{d}^i(\theta_f, \theta_d), d^i), \quad (7.1)$$

where $\hat{d}^i$ is the probability of the i th example to belong to the target domain and N is the number of examples for each domain (i.e., there are $2N$ examples in total). In this loss, L_{seg} is the dice loss and encourages both the feature extractor and the label predictor to learn features useful for source domain image segmentation. The expression L_{dom} is the cross-entropy and encourages (i) the feature extractor to learn domain-invariant features and (ii) the domain classifier to predict whether the features in Lz are activated by a source or target domain input.

$$L_{\text{tot}}(\theta_f, \theta_y, \theta_d) = \sum_{\substack{i=1..N \\ d^i=0}} L_{\text{seg}}(\hat{\mathbf{Y}}^i(\theta_f, \theta_y), \mathbf{Y}^i) + \sum_{i=1..N} L_{\text{dom}}(\hat{d}^i(\theta_f, \theta_d), d^i), \quad (7.2)$$

The label predictor's output is:

$$\hat{\mathbf{Y}}^i(\theta_f, \theta_y) = G_y(G_f(\mathbf{X}^i; \theta_f); \theta_y), \quad (7.3)$$

where $\mathbf{X}^i$ is the i th input image. The saddle point of (7.2) is estimated by backpropagation. To control the strength of the interaction between the feature extractor and the domain classifier, we follow [53] and use the *gradient reversal layer*, defined as $R_\lambda(\mathbf{a}) = \mathbf{a}$ and $dR_\lambda(\mathbf{a})/d\mathbf{a} = -\lambda\mathbf{I}$. The larger λ , the higher the interaction between domain classifier and feature extractor. The domain classifier's output then is written as:

$$\hat{d}^i(\theta_f, \theta_d) = G_d(R_\lambda(G_f(\mathbf{X}^i; \theta_f)); \theta_d). \quad (7.4)$$

In practice, setting λ to a fixed value throughout learning leads to instabilities. To address this issue, we set λ to zero during the first e_0 epochs (i.e., the feature extractor and domain classifier learn their tasks independently) and then increase λ linearly until it reaches the value λ_{max} after the total number of epochs n_{epochs} :

$$\lambda(e) = \max \left(0, \lambda_{\max} \frac{e - e_0}{n_{\text{epochs}} - e_0} \right), \quad (7.5)$$

where e is the current epoch. In this expression, $\lambda_{\max}$, e_0 and n_{epochs} are hyper-parameters.

In supervised learning, hyper-parameters are most often chosen by comparing the performance on a validation set for different values of hyper-parameters. When the training involves images from different distributions, the validation set must contain images from the distribution on which we want the system to perform best (i.e., validation set contains annotated images from the target domain). In an unsupervised domain adaptation setting, however, we need alternative ways to choose hyper-parameters (since annotations are assumed not to be available for the target domain). In our application, we assess the performance of our method by observing (i) the DSC on the test source data and (ii) the domain classifier accuracy (i.e., the proportion of images correctly classified as CT or CBCT). More specifically, we observe good performance on the target domain when the following conditions¹ are met:

- during the first e_0 epochs, (i.e., when the feature extractor and domain classifier are trained independently), the test domain classifier accuracy steadily reaches a test accuracy close to its theoretical maximum (i.e., its value at convergence when $\lambda = 0$ throughout training);
- during the remaining training, the test domain classifier accuracy decreases to 0.5 and;
- during the whole training, the DSCs of the bladder, rectum, and prostate on the source domain reach values close to their theoretical maxima (its value at convergence when there is no UDA).

We selected hyper-parameter values that best meet these criteria. For the feature extractor and label predictor, our network starts with 16 feature maps in the first level (corresponding to L1) and is doubled in

¹Note that these conditions do not require annotated target images, so that hyper-parameters can be selected using only annotated source images and non-annotated target images. In this study, the labeled CBCTs (30 images from cohort $\mathcal{C}_4$) are used only to report the method's performance.

each level. For the domain classifier, the number of feature maps in the first level is the number of feature maps in Lz divided by two, and then it doubles with each increase in level. In the example in Fig. 7.2, where the domain classifier is connected on L9 (which has 64 feature maps), the domain classifier has 32 feature maps in its first level, followed by 64, 128, and 256 in the following levels. The number of units in the last fully connected layer is always equal to 256. Due to memory limitation, we choose a batch size of 2. Our model is optimized with Adam and a learning rate of 10^{-4} . For the trade-off parameter schedule, a value of $e_0 = 50$ was chosen and $\lambda_{\max}$ was set to 0.01, 0.03, 0.1 or 0.3 depending on the layer adapted. All networks were trained for 150 epochs.

The source code is publicly available on
<https://github.com/eliottbrion/unsupervised-domain-adaptation-unet-keras>.

7.3.3 Intensity-based data augmentation

In the context of deep learning, data augmentation most often refers to the application of "geometric" transformations to images (i.e., small rotations, translations, and shear) in the training set in order to improve generalization on the test set. The rationale is that samples from the training and test sets are all generated by the same distribution, and the goal of geometric data augmentation is to mitigate the small number of training samples. However, when considering domain adaptation, training and test samples belong to different distributions. Hence, on top of the geometric data augmentation, we have performed an "intensity-based" data augmentation. The goal is to bring the source distribution closer to the target distribution. The following augmentations have been implemented:

- **Brightness:** a random offset is added to the image: $\mathbf{X} \leftarrow \mathbf{X} + u\mathbf{J}$, where $u \sim U[-100, 100]$ and $\mathbf{J} \in \mathbb{R}^{160 \times 160 \times 128}$ is a tensor with all entries equal to one.
- **Contrast:** the mean separation between low- and high-intensity voxels of the image is randomly modified: $\mathbf{X} \leftarrow u(\mathbf{X} - \bar{X}\mathbf{J}) + \bar{X}\mathbf{J}$.
- **Sharpness:** high-frequency detail is randomly strengthened or weakened: $\mathbf{H} = \mathbf{X} - G(\mathbf{X}, 2^2)$ and $\mathbf{X} \leftarrow \mathbf{X} + u(\mathbf{H} - \bar{H}\mathbf{J})$, where

$\mathbf{H}$ is a high-frequency image obtained by subtracting a Gaussian blurred (with standard deviation of 2 HU) version of the image and $u \sim U[-0.2, 1.8]$.

- **Noise:** random Gaussian noise with zero mean and standard deviation 20 HU is added to each voxel of the image: for $1 \leq i, j \leq 160$ and $1 \leq k \leq 128$, $X_{i,j,k} \leftarrow X_{i,j,k} + v_{i,j,k}$, where $v_{i,j,k} \sim N(0, 20^2)$.
- **Linear:** we assume unknown linear correspondence between CT and CBCT intensities: $\mathbf{X} \leftarrow m\mathbf{X} + p\mathbf{J}$, where $m \sim U[0.5, 1.5]$ and $p \sim U[-50, 50]$.
- **Bilinear:** the correspondence between CT and CBCT intensities are often best modelled by a piecewise linear function instead of a strictly linear function. This hinged line is inspired by typical *calibration curves* from Hounsfield units to other physical quantities. We model this correspondence during training: for $1 \leq i, j \leq 160$ and $1 \leq k \leq 128$,

$$X_{i,j,k} \leftarrow \begin{cases} m_1 X_{i,j,k} + p, & \text{if } X_{i,j,k} \leq 0 \\ m_2 X_{i,j,k} + p, & \text{otherwise} \end{cases}$$

where $m_1, m_2 \sim U[0.5, 1.5]$ and $p \sim U[-50, 50]$.

To finish, the influence of the data size on the accuracy of the model is also studied by increasing the training size from 30 to 74.

7.3.4 Performance metrics and comparison baselines

Performance metrics. The performances of the different methods are measured with one overlap-based metric (the dice similarity coefficient, or DSC) and one distance-based metric (the symmetric mean boundary distance, or SMBD). They are respectively defined by:

$$\text{DSC} = \frac{2|M \cap P|}{|M| + |P|}, \quad (7.6)$$

$$\text{SMBD} = \frac{\overline{D}(M, P) + \overline{D}(P, M)}{2}, \quad (7.7)$$

where M and P are the sets containing the matricial indices of the manual and predicted segmentation 3D binary masks (respectively), $\overline{D}(M, P)$ is the mean of $D(M, P)$ over the voxels of Ω_M , and $D(M, P) = \{\min_{x \in \Omega_P} \|\mathbf{s} \odot (\mathbf{x} - \mathbf{y})\|, \mathbf{y} \in \Omega_M\}$, where Ω_M, Ω_P are respectively the boundaries extracted from M and P and $\mathbf{s}^\top = (1.2, 1.2, 1.5)$ is the pixel spacing in mm.

Deep learning baselines. A naive strategy for learning to segment target domain images from a dataset of labeled source domain images would be simply to train a neural network on those source images (CTs from cohort $\mathcal{C}_1$, $\mathcal{C}_2$, or $\mathcal{C}_5$, depending on the experiment) and then test it on test domain images (always the CBCTs from cohort $\mathcal{C}_4$). The CBCTs from cohort $\mathcal{C}_3$ are not used for tests since the images are already used for training the adversarial networks and we wish to report the performance on an independent test set. This "source only" strategy is a valid baseline and provides a lower bound on what can be achieved with adversarial networks or intensity-based data augmentation. As a second baseline, we (i) produce artificial CBCTs by passing CTs from cohort $\mathcal{C}_1$ through a cycleGAN [180] and (ii) use these artificial CBCTs to train a U-Net segmentation network. Code from Keras (<https://keras.io/examples/generative/cyclegan/>) has been adapted to train the 2D cycleGAN slice-wise on CTs from cohort $\mathcal{C}_1$ and CBCTs from cohort $\mathcal{C}_3$ for image translation. Finally, we compared our methods with PSIGAN [77], a UDA approach modeling the co-dependency between images and their segmentation as a joint probability distribution.

Registration baselines. In the context of CBCT segmentation for radiotherapy, the contoured planning CT scan is an important available prior. Hence, using deformable image registration (DIR), we can map the contours of the planning CT scan to the daily CBCT scan. Different mapping rules define different registration algorithms, and in this paper, three registration algorithms are compared: rigid, the registration method implemented in RayStation (a commercial treatment planning system), and Morphons. Rigid and RayStation are parametric methods: they look for the parameters of a deformations model – either linear (rigid) or non-linear (RayStation) – that yield the best match between the planning CT and daily CT scan. The diffeomorphic

morphons’ DIR algorithm, a non-parametric method, is implemented in OpenReggui (<https://openreggui.org/>) [74]. This method exploits the local phase of the scans to perform the registration. Therefore, it is suited for the registration of scans with different contrast enhancement, such as CT and CBCT scans. The diffeomorphic version of the algorithm forces anatomically plausible deformations. The registration has been performed between the CT and CBCT scans of patient cohort $\mathcal{C}_4$.

7.4 Results

In this section, we investigate three different strategies for improving cross-domain generalization of deep neural networks: adversarial networks, intensity-based data augmentation, and intensity-based data augmentation with a larger training set.

7.4.1 Adversarial networks

We train our adversarial networks with adaptations on L3, L6, L9, and L11, using the 30 CTs from cohort $\mathcal{C}_1$ and the 30 CBCTs from cohort $\mathcal{C}_3$. The CTs from cohort $\mathcal{C}_2$ are used to monitor the performance of the segmenter on source domain images. The performance of the generated models on CBCTs from $\mathcal{C}_4$ is compared with baseline methods (training on source and DIR algorithms) in Table 7.1. Compared with “source only” training, adversarial networks do not bring improvement when the adaptation is made on L3. An explanation is that since this layer corresponds to low-level features, discriminating between CT and CBCT does not make sense (detecting edges is relevant for both CT and CBCT). When the adaptation is performed on L6, a small improvement is observed for the bladder, while the performance for the prostate decreases slightly. When adapted on L9, the performance is similar for the bladder and prostate and slightly better for the rectum. Finally, when we adapt on L11 (i.e., the pre-softmax activations), the model performance is improved for all three organs, both in term of DSC and SMBD. This means that adversarial learning forces the features learned on CT to align with CBCTs so that they generalize to CBCT better. The adaptation proves especially useful for the rectum, which experiences the most dramatic change in intensities between the source and target domain images (see

histograms in Fig. 7.3). Although slightly outperforming the rigid and RayStation (RS) intensity-based registration baselines (similar DSC and smaller SMBD) for the bladder, the performance of adversarial networks (L11) remains below these algorithms for the rectum and prostate, and below the morphons registration baseline for all three organs.

Table 7.1: Performance (mean $\pm$ std) on the 30 CBCTs from cohort $\mathcal{C}_4$ for adversarial networks and for the baseline methods (train on source and DIRs). DSC: Dice similarity coefficient, SMBD: symmetric mean boundary distance, DL: deep learning, DIR: deformable image registration, RS: RayStation. $^\diamond$ For at least one patient, the model predicts no segmentation at all for the organ and therefore the SMBD of that patient (as well as the mean over all patients) is not defined.

Method	DSC			SMBD (mm)		
	Bladder	Rectum	Prostate	Bladder	Rectum	Prostate
DIR, Rigid image registration	0.757 $\pm$ 0.150	0.629 $\pm$ 0.087	0.721 $\pm$ 0.130	5.48 $\pm$ 3.53	5.77$\pm$1.96	3.90 $\pm$ 1.67
DIR, RS intensity-based	0.775 $\pm$ 0.154	0.640 $\pm$ 0.096	0.720 $\pm$ 0.133	5.01 $\pm$ 3.50	5.61 $\pm$ 2.12	3.91 $\pm$ 1.74
DIR, Morphons	0.813$\pm$0.154	0.653$\pm$0.169	0.731$\pm$0.146	4.00$\pm$3.17	5.78 $\pm$ 4.08	3.63$\pm$1.83
DL, Train on source	0.749 $\pm$ 0.212	0.179 $\pm$ 0.241	0.629 $\pm$ 0.169	5.76 $\pm$ 5.96	Undefined $^\diamond$	4.96 $\pm$ 2.29
DL, Adversarial (L3)	0.764 $\pm$ 0.180	0.168 $\pm$ 0.243	0.636 $\pm$ 0.137	5.63 $\pm$ 5.21	Undefined $^\diamond$	5.04 $\pm$ 2.19
DL, Adversarial (L6)	0.771 $\pm$ 0.164	0.193 $\pm$ 0.263	0.609 $\pm$ 0.145	4.88 $\pm$ 3.48	Undefined $^\diamond$	5.41 $\pm$ 2.08
DL, Adversarial (L9)	0.735 $\pm$ 0.192	0.210 $\pm$ 0.261	0.624 $\pm$ 0.158	4.98 $\pm$ 3.50	Undefined $^\diamond$	5.26 $\pm$ 2.42
DL, Adversarial (L11)	0.787 $\pm$ 0.166	0.447 $\pm$ 0.181	0.660 $\pm$ 0.158	4.60 $\pm$ 3.74	8.04 $\pm$ 3.13	4.72 $\pm$ 2.40

7.4.2 Intensity-based data augmentation

Comparison of different intensity-based data augmentation types

While adversarial networks align the features activated by source and target images, an alternative approach consists in altering the source images to align them with the target images. The performance of different intensity-based data augmentation strategies is compared in Table 7.2 using standard (non adversarial) training. In these experiments, we keep the same CTs (from cohort $\mathcal{C}_1$) as in the previous section for training and the same CBCTs (from cohort $\mathcal{C}_4$) for testing. Contrary to the previous section, images from cohorts $\mathcal{C}_2$ and $\mathcal{C}_3$ are not used. Strategies involving modification in levels of contrast, sharpness and noise are inefficient and perform similarly or worse than the "source only" baseline. Brightness, linear, and bilinear data augmentations, however, outperform the

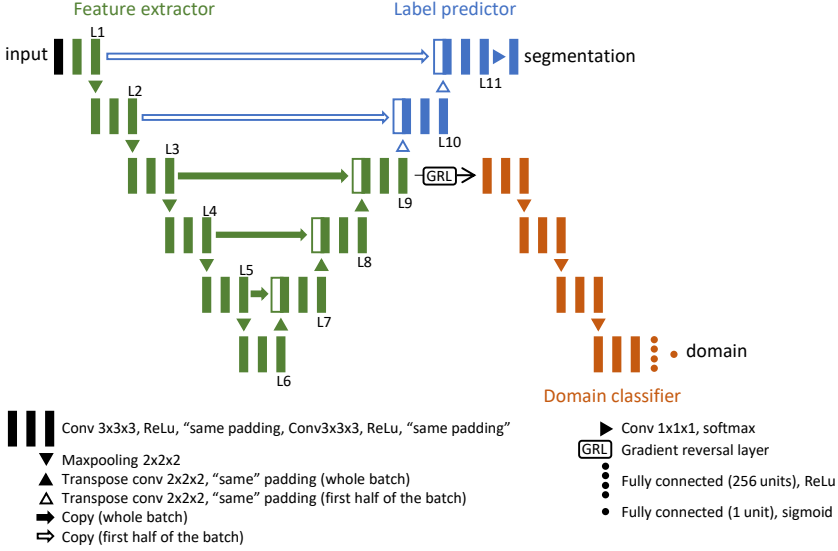


Figure 7.2: Our proposed adversarial architecture adapts 3D U-Net to unsupervised domain adaptation by backpropagation (in this figure, the layer L_z , $z = 9$ is adapted). The goal is to learn, from labeled source examples (in our case, CTs), features that are also useful for segmenting images from a different yet similar target domain (CBCTs). U-Net is split in two parts: a feature extractor and a label predictor. The feature extractor learns features from the whole batch (both source and target), with its final layer L_z to (i) be useful for the label predictor and (ii) fool the domain classifier. The label predictor learns feature from half the batch in the layer L_z (the source examples) aiming to predict a segmentation mask for these source examples. The domain classifier uses these same features L_z to classify all images in the batch (both source and target) as being either source or target.

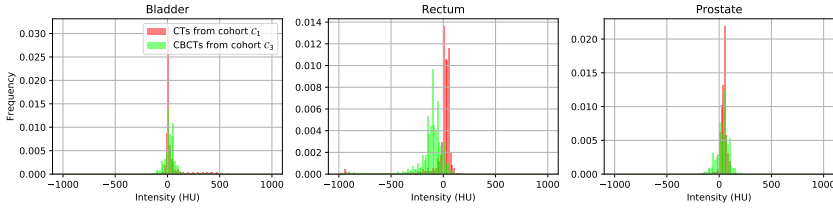


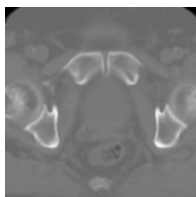
Figure 7.3: Histograms differ significantly between the 30 CTs from cohort $\mathcal{C}_1$ and the 30 cone beam CTs from cohort $\mathcal{C}_3$ (the two cohorts used for training the adversarial networks), especially inside the rectum.

”source only” baseline on all three organs, in terms both of DSC and SMBD. Brightness and linear transformations perform similarly for all three organs, with slightly worse performance than morphons for the bladder, worse performance for the prostate, and better performance for the rectum. The bilinear method, finally, performs similarly to morphons for the bladder (larger DSC, larger SMBD) and worse on both the rectum and prostate. CycleGAN, when compared to the contrast, sharpness and noise transformations, performs better on some organs and worse on others. In contrary, the brightness, linear, and bilinear transformations all outperform cycleGAN. One possible explanation for the relative weakness of cycleGAN is that it generates blurry images. Blur is problematic because it decreases the contrast at organs boundaries, the most challenging areas for segmentation. An example of a slice of an original CT, as well as its transformation through cycleGAN and brightness operation is depicted in Fig. 7.4.

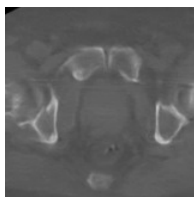
Since adversarial networks and brightness-based data augmentation both yield improvements over the baseline ”source only” method, we investigate the performance of the combined approaches. We chose brightness over other intensity-based data augmentation strategies because it has the best balance of performance over the three organs. Combining both approaches led to no improvement compared with brightness augmentation alone (see last row in Table 7.2).

Table 7.2: Performance (mean $\pm$ std) on the 30 CBCTs from cohort $\mathcal{C}_4$ for intensity-based data augmentation and for the baseline methods (training on source and DIRs). DSC: dice similarity coefficient, SMBD: symmetric mean boundary distance, DL: deep learning, DIR: deformable image registration, RS: RayStation, Adv.: adversarial networks. $^\diamond$ For at least one patient, the model predicts no segmentation at all for the organ and therefore the SMBD of that patient (as well as the mean over all patients) is not defined.

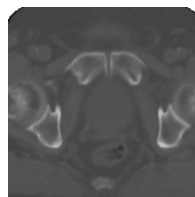
Method	DSC			SMBD (mm)		
	Bladder	Rectum	Prostate	Bladder	Rectum	Prostate
DIR, Morphons	0.813 $\pm$ 0.154	0.653 $\pm$ 0.169	0.731$\pm$0.146	4.00$\pm$3.17	5.78 $\pm$ 4.08	3.63$\pm$1.83
DL, Train on source	0.749 $\pm$ 0.212	0.179 $\pm$ 0.241	0.629 $\pm$ 0.169	5.76 $\pm$ 5.96	Undefined $^\diamond$	4.96 $\pm$ 2.29
DL, CycleGAN	0.744 $\pm$ 0.140	0.602 $\pm$ 0.119	0.521 $\pm$ 0.146	5.77 $\pm$ 3.39	5.69 $\pm$ 2.13	6.86 $\pm$ 2.32
DL, Brightness	0.791 $\pm$ 0.191	0.690 $\pm$ 0.121	0.658 $\pm$ 0.135	4.38 $\pm$ 4.08	4.78$\pm$2.03	5.08 $\pm$ 3.08
DL, Contrast	0.765 $\pm$ 0.179	0.205 $\pm$ 0.247	0.618 $\pm$ 0.180	6.46 $\pm$ 5.20	Undefined $^\diamond$	5.47 $\pm$ 2.96
DL, Sharpness	0.661 $\pm$ 0.164	0.183 $\pm$ 0.252	0.584 $\pm$ 0.182	6.48 $\pm$ 3.04	Undefined $^\diamond$	5.70 $\pm$ 2.51
DL, Noise	0.672 $\pm$ 0.173	0.158 $\pm$ 0.234	0.534 $\pm$ 0.199	6.91 $\pm$ 4.21	35.06 $\pm$ 26.99	8.96 $\pm$ 5.39
DL, Linear	0.775 $\pm$ 0.125	0.694$\pm$0.100	0.651 $\pm$ 0.133	4.70 $\pm$ 2.85	5.11 $\pm$ 2.37	5.07 $\pm$ 2.63
DL, Bilinear	0.837$\pm$0.107	0.562 $\pm$ 0.174	0.666 $\pm$ 0.117	4.67 $\pm$ 3.60	6.13 $\pm$ 2.67	4.77 $\pm$ 2.42
DL, Brightness+adv.	0.777 $\pm$ 0.157	0.694$\pm$0.097	0.654 $\pm$ 0.137	4.83 $\pm$ 3.72	4.78$\pm$1.92	5.08 $\pm$ 2.52



(a) An original CT slice.



(b) CT transformed with cycleGAN.



(c) CT transformed with brightness.

Figure 7.4: Example of intensity-based data augmentations.

Brightness with the extended source dataset

In Section 7.4.2, we used only 30 training CTs (from cohort $\mathcal{C}_1$) in order to make a fair comparison with adversarial networks. As a last experiment, we investigated whether using all available CTs (74 examples for cohorts $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_5$) improved the performance. Table 7.3 shows that the proposed method performs as well as morphons for the prostate, and outperforms it on the bladder and rectum. Actual segmentations on a representative patient are shown in Fig. 7.5. It also outperforms adversarial networks (L11) on all three organs.

Table 7.3: Performance (mean $\pm$ std) on the 30 CBCTs from cohort $\mathcal{C}_4$ for adversarial networks, intensity-based data augmentation, baseline methods as well as the performance of previous works on other datasets. $\diamond$ For at least one patient, the model predicts no segmentation at all for the organ and therefore the SMBD of that patient (as well as the mean over all patients) is not defined. $\dagger$ Results reported on a test set containing both CBCT and CT scans. $\ddagger$ The authors computed the root mean squared boundary distance rather than the SMBD. $\S$ The authors computed the mean boundary distance and not the SMBD. For deep learning methods, n_{CBCT} and n_{CT} are the number of annotated CBCTs and CTs used for training (respectively).

Study	Method	DSC			SMBD (mm)		
		Bladder	Rectum	Prostate	Bladder	Rectum	Prostate
Ours	DIR, Morphons	0.813 $\pm$ 0.154	0.653 $\pm$ 0.169	0.731 $\pm$ 0.146	4.00 $\pm$ 3.17	5.78 $\pm$ 4.08	3.63$\pm$1.83
Ours	DL, Train on source ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 30$)	0.749 $\pm$ 0.212	0.179 $\pm$ 0.241	0.629 $\pm$ 0.169	5.76 $\pm$ 5.96	Undefined $^\diamond$	4.96 $\pm$ 2.29
Ours	DL, Adversarial (L11) ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 30$)	0.787 $\pm$ 0.166	0.447 $\pm$ 0.181	0.660 $\pm$ 0.158	4.60 $\pm$ 3.74	8.04 $\pm$ 3.13	4.72 $\pm$ 2.40
Ours	DL, Train on source ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 74$)	0.775 $\pm$ 0.138	0.173 $\pm$ 0.235	0.628 $\pm$ 0.145	5.11 $\pm$ 3.25	Undefined $^\diamond$	6.65 $\pm$ 3.78
Ours	DL, Brightness ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 74$)	0.837 $\pm$ 0.142	0.701 $\pm$ 0.114	0.734$\pm$0.113	3.55 $\pm$ 2.44	4.80 $\pm$ 2.13	3.63$\pm$1.79
Ours	DL, PSIGAN ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 30$)	0.825 $\pm$ 0.113	0.747 $\pm$ 0.083	0.674 $\pm$ 0.102	3.74 $\pm$ 2.11	3.10 $\pm$ 1.19	4.16 $\pm$ 1.47
Ours	DL, PSIGAN ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 74$)	0.868$\pm$0.094	0.762$\pm$0.093	0.719 $\pm$ 0.083	2.90$\pm$2.09	2.96$\pm$1.23	3.66 $\pm$ 1.36
Ours	DL, PSIGAN+Brightness ($n_{\text{CBCT}} = 0, n_{\text{CT}} = 74$)	0.850 $\pm$ 0.110	0.697 $\pm$ 0.140	0.685 $\pm$ 0.101	3.98 $\pm$ 2.86	3.62 $\pm$ 1.69	3.90 $\pm$ 1.66
[52]	DL ($n_{\text{CBCT}} = 80, n_{\text{CT}} = 0$)	0.96$\pm$0.03	0.93$\pm$0.04	0.91$\pm$0.08	0.65$\pm$0.67§	0.72$\pm$0.61§	0.93$\pm$0.96§
[101]	DL ($n_{\text{CBCT}} = 42, n_{\text{CT}} = 74$)	0.874 $\pm$ 0.096	0.814 $\pm$ 0.055	0.758 $\pm$ 0.101	2.47 $\pm$ 1.93	2.38 $\pm$ 0.98	3.08 $\pm$ 1.48
[145]	DL ($n_{\text{CBCT}} = 300, n_{\text{CT}} = 300$)	0.932 †	0.871 †	0.840 †	2.57 $\pm$ 0.54 $^{\dagger, \ddagger}$	2.47 $\pm$ 0.64 $^{\dagger, \ddagger}$	2.34 $\pm$ 0.68 $^{\dagger, \ddagger}$
[62]	DL ($n_{\text{CBCT}} = 124, n_{\text{CT}} = 88$)	0.88	0.71	-	-	-	-
[123]	DIR, MIM intensity-based	$\sim$ 0.80	$\sim$ 0.40	$\sim$ 0.55	-	-	-
[123]	DIR, RS intensity-based	$\sim$ 0.78	$\sim$ 0.70	$\sim$ 0.75	-	-	-
[153]	DIR, RS intensity-based	0.69 $\pm$ 0.07	0.75 $\pm$ 0.05	0.84 $\pm$ 0.05	-	-	-
[168]	DIR, Cascade MI-based	$\sim$ 0.83	$\sim$ 0.77	$\sim$ 0.80	$\sim$ 2.6 §	$\sim$ 2.3 §	$\sim$ 2.3 §
[90]	DIR, Rigid on bone and prostate	0.85 $\pm$ 0.05	-	0.82 $\pm$ 0.04	-	-	-
[156]	DIR, Denons	0.73	-	0.80	-	-	-
[162]	Patient specific model	$\sim$ 0.87	-	-	-	-	-
[24]	Patient specific model	0.78	-	-	-	-	-

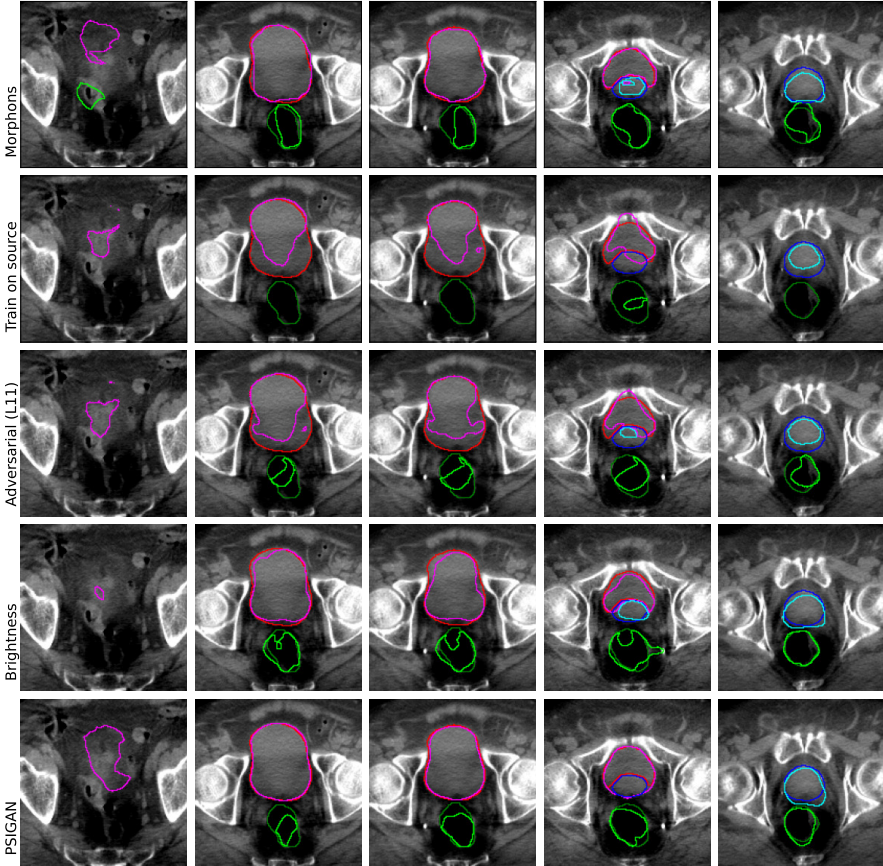


Figure 7.5: Comparison of morphons, "train on source", adversarial networks with adaptation on L11, brightness, and PSIGAN segmentation for a representative patient from cohort $\mathcal{C}_4$. Each column corresponds to a slice of the same CBCT scan. Dark colors represent reference segmentations, while light colors show predictions from the different algorithms. Compared to "train on source", Adversarial networks (L11) brings small improvements for the bladder (by detecting slightly larger portions of it) and the prostate (it is missed by "train on source" in the fourth column), as well as large improvement in rectum (since it is completely missed by "source on train" on most slices). Brightness performs similar to morphons for the prostate, slightly outperforms it for the bladder (fewer false negatives in the first column) and performs better for the rectum (no false negatives in the first column and more accurate delineation in the fifth). Color should be used for this figure in print.

7.5 Discussion

U-Net trained on CTs for bladder, rectum and prostate segmentation does not generalize well on CBCTs, as shown in Table 7.3. For the rectum, in particular, it produces many false negatives (this is illustrated in Fig. 7.5). This may be explained by the fact that deep learning learns correlations between patterns of intensities and presence/absence of a given organ, without prior information about anatomy. For instance, a network trained in a non-adversarial way has no incentive to avoid giving a zero prediction for the rectum, even if every patient has one. Image registration methods, in contrast, deform the anatomy from planning to the daily image. We show that adversarial networks applying adaptation of the pre-softmax activations (L11) yield minor improvements (compared to source-only training) on the bladder and prostate, and significant improvement on the rectum (for this last organ, the DSC more than doubles). For this organ, the improvement can be attributed to the introduction of an anatomical a priori. By encouraging the network to have indistinguishable pre-softmax activations between CT and CBCT, UDA forces the network to produce contours for the rectum, even if the intensities in the test CBCTs are not similar to the ones observed in the training CTs.

In the context of radiotherapy, one disposes of an interesting a priori information to segment the CBCT, namely, the planning CT. Registration takes advantage of this information. This algorithm is a relevant baseline because, similar to the two methods proposed in this paper (adversarial networks and intensity-based data augmentation), it does not have to be trained on annotated CBCTs. Adversarial networks' performance compared to registration's is contrasted. We applied three registration algorithms to our dataset. Adversarial networks perform better than rigid registration and similar to the RayStation commercial software on the bladder and perform worse on the two other organs. Compared with the morphons algorithm, adversarial networks (irrespective of the layer adapted) perform worse on all three organs. We also compared our results to other works on other datasets. For the bladder, the comparison is mixed, with adversarial networks (L11) performing worse [90, 168], similarly [123] or better [153, 156], depending on the study. For the rectum and prostate, adversarial networks (L11) per-

form worse than most other authors' methods, with the exception of MIM intensity-based [123]. Note that those studies report results on a dataset different from ours. It is therefore difficult to determine what part of the difference in performance can be attributed to the method used, and what part to differences in the degree of difficulty in the test datasets. A limitation of the comparison shown in Table 7.3 is that it does not compare our methods with other unsupervised domain adaptation frameworks on our data.

An advantage of adversarial networks over registration is faster inference. For each new test image, registration requires iterative image alignment with gradient descent, which takes several minutes. This prevents its application in the context of adaptive treatment, when the patient lies on the couch while treatment adaptation is being computed. In contrast, while adversarial networks require long (a few hours) training times, prediction does not entail iterative optimization but only matrix multiplication, which takes less than a second.

For the bladder, we compared adversarial networks with two patient-specific models. Here the conclusion is also mitigated, with adversarial networks (L11) performing worse than one [24] and better than the other [162]. A drawback of patient specific-models is that they require contouring several images of the target patient by hand, which is long, tedious, and therefore often not practical. This is not necessary for adversarial networks.

While the performance of adversarial networks compared with registration and patient-specific models is contrasted, this method is interesting since, to our knowledge, this is the first time that UDA has been applied to segmentation using U-Net with two settings: long skip connections and fully 3D. Previous work chose to remove the long skip connections when performing UDA, probably out of the concern that the network could use them to "bypass" the features alignment constraint. Analyzing the impact of skip connections for adversarial networks in UDA settings remains to be addressed and would be interesting to investigate in future studies. A second interesting feature of our approach is that it is built on a 3D U-Net instead of aggregating the outputs from a 2D U-Net, which is what is most often done with UDA in segmentation. Using 3D leverages useful information from an additional dimension. A possible worry about using 3D instead of 2D is that it artificially re-

duces the number of different samples seen by the discriminator (by a factor equal to the number of slices per image). Again, although the comparison between 2D and 3D is beyond the scope of this study, we have shown that training a 3D network is possible, even if few samples are available for training.

A challenge with adversarial networks is that their training is notoriously unstable. In this paper, we also analyzed an alternative method where input images are transformed so that they look more like CBCTs. The method with best average performance across organs was brightness-based augmentation, which consists in adding a random offset on the training images. For fair comparison with adversarial networks, a model with brightness augmentation was first trained on 30 CTs, performing similarly to adversarial networks (L11) for the prostate and better on the bladder and rectum. When training is performed on a larger set of 74 CTs, this method performs similarly to morphons on prostate and better on the two other organs. Compared with previous studies based on DIR, brightness with the extended dataset performs generally similarly or better on the bladder. For the rectum and the prostate, however, our algorithm performs equally well or worse (it is better than MIM intensity-based from [123] only). To summarize this comparison between intensity-based data augmentation and previous work using other strategies, our method thus has relevance for the bladder, as well as for applications where short inference time matters, such as adaptive radiotherapy.

Table 7.3 summarizes our better results, and presents the results obtained from DIR methods, as well as DL methods trained fully on CBCT data or a combination of both CT and CBCT. Our results cannot be directly compared with these DL methods due to the inherent challenge of training with unlabeled data, which is the bottleneck of current clinical practice. In addition, the number of patients used in these studies is often larger than our database (Schreier et al., for example, used 300 patients [145]). Acquiring databases of patients larger than 50-100 also represents an added complexity for small to medium-sized hospitals. Given the small database and the challenge of training with unlabeled CBCTs, our method achieves very good results: a max. difference of only 0.1-0.2 in the dice for the best reported results in the literature. In addition, the results in Section 4.3 suggest that increasing the size of

the database for adversarial training, too, may lead to an improvement of the results. The PSIGAN method with 74 training CTs outperforms the brightness-based data augmentation on the bladder and rectum but performs worse on the prostate. This suggests that modeling the joint distribution of images and their segmentation can be useful for CT to CBCT adaptation. We investigated adding brightness-based data augmentation on top of PSIGAN, but it hurt performance. Similarly, we saw in Section 7.4.2 that adding brightness-based data augmentation on top of gradient reversal layer networks decreased performance for the bladder and prostate, even it brought improvement on the rectum. Eventually, brightness-based data augmentation is an interesting technique that is simple and works well alone, but with ambiguous effects when combined with other UDA methods.

We did not use batch normalization since it hurted performance. A similar situation was observed in the original 3D U-Net paper [31] for fully-automated segmentation. A possible explanation is that there is too little variety in the training batches with our training set of only 30 images. For this reason, methods relying on batch normalization for domain adaptation [105] cannot be used in our situation.

7.6 Conclusions

In this study, we propose an extension of 3D U-Net on unsupervised domain adaptation, applying adversarial networks to the segmentation of organs (bladder, rectum and prostate) in cone beam CT. For a model trained on CT, this method improves generalization to CBCTs, using only non-annotated CBCTs. Our code is made public and can be used for any application experiencing a shift of distribution between training and test images, a recurring problem in biomedical imaging. Although less accurate than morphons, a registration-based algorithm, this method is faster (less than one second versus several minutes). This is essential for online treatment adaptation. We also show that when intensity-based data augmentations are applied to training CTs, the network generalizes better to CBCT, reaching a level similar to (prostate) or better (bladder, rectum) than the best registration-based algorithms. Intensity-based data augmentation is best used alone since it has ambiguous effects when combined with other UDA methods. In the future,

this could be used in radiotherapy treatment to measure and ultimately reduce the dose delivered to healthy organs, while delivering the prescribed dose to the target.

Chapter 8

Conclusion

In this thesis we have addressed two major challenges for radiotherapy, a cancer treatment aimed at irradiating tumors while preserving healthy organs, namely, (i) tedious manual contouring that slows the treatment planning and (ii) poorly monitored anatomical variations across treatment sessions that hamper dose conformity. We have shown that deep learning models can mitigate these two issues if they are given access to data from multiple hospitals, yet centralizing data raises technical and legal challenges.

In the course of this thesis, we established contacts with multiple radiotherapy clinics. After convincing the heads of their radiotherapy departments and their ethics committees of the merits of our undertaking, we reached an agreement on a legal contract with four of them. We led the collection and annotation of a database of 402 images from two clinical partners. Yet the fact that we could not convince some radiotherapy clinics to work with us, as well as the fact that we had to abandon one of them for technical reasons and another due to time constraints, illustrates the limitations of the data centralization approach.

To address this issue, we proposed a distributed learning framework based on federated Byzantine agreement and model encryption. A use-case based on bladder segmentation on CTs proved the method’s feasibility.

A neural network for contour propagation was then developed. This deep learning framework was shown to improve the network’s performance in contouring a given slice when a neighboring contour was given

as prior, thereby accelerating the treatment contouring while preserving accuracy.

Finally, we have shown how large databases of contoured CTs can be used to train a network for the bladder, rectum, and prostate segmentation on Cone Beam CT, a modality for daily imaging. Although this method improves treatment accuracy by tracking structures of interest throughout treatment sessions, its implementation still requires the tedious contouring of a few CBCTs manually. To overcome this constraint, we propose two methods, one built on domain adversarial networks and the other on intensity-based data augmentation, to train a network to contour CBCTs using only contoured CTs and non-contoured CBCTs.

Limitations

We have determined the application domain of our proposed methods. Our contour propagation network is helpful mainly at given interslice distances, outside of which it performs similarly to other methods. Cross-domain data augmentation is applicable only when a few labeled CBCTs are available, which is not always the case. While our two other proposed methods — domain adversarial networks and intensity-based data augmentation — do not suffer from this limitation of application range, they perform less well. More generally, for all proposed algorithms, a gap in performance compared with human experts remains.

Our models were trained and validated on specific scanners and with specific labels. They are not expected to perform well on other scanners or images with artifacts, and they cannot produce contours conforming to other contouring protocols (e.g., the rectal wall only instead of the whole rectum). Moreover, the predicted prostate contours are useful only when the CTV corresponds to the prostate itself, which excludes many patients (e.g., those from whom this organ has been removed). Our method also assumes that a CBCT is available for daily imaging, while other image modalities (such as X-rays) are sometimes used instead.

Some restrictions in terms of data and methodology limit the scope of our findings. In many instances, small differences in performance are compared over small (fewer than 100 cases) samples. In experiments related to contour propagation, domain adversarial networks,

and intensity-based data augmentation, no statistical test has been performed to support claims of difference in performance. Even for the experiments where these tests have been used (related to cross-domain data augmentation), the small size of the test set does not guarantee that similar average performance would be observed if the algorithm were deployed in practice.

Even if more data were available, lack of explainability of deep neural networks (compared with humans' expertise, or even with registration and patient-specific models) could slow down their adoption in practice. Deciding where the dose is deposited in someone's body is an important decision, and not everyone is ready to let an algorithm that we do not fully understand make such a decision. However, one can argue that this is not a big deal as long as a randomized trial has demonstrated the automated method to have clinical value. The mechanisms underlying many drugs' actions remain unknown, yet most people accept to use them after trials have proven their efficacy.

A machine learning algorithm can only be as good as the data provided for its training. Unfortunately, there is no way to know the exact position of an organ in an image. First, because in some cases the information is not present in the image, for example when there is poor contrast between bladder and prostate on CBCT. Moreover, while the contours determined by doctors provide the best approximation, they are subject to inter- and intra-observer variabilities [50]. Also, we evaluated the algorithm's performance with mathematical metrics. In practice, a given metric value can correspond to either a good or bad contour, depending on where the errors occur.

Finally, there are the limitations of radiotherapy itself. In particular, there are doubts about its efficacy compared with active monitoring for prostate cancer [63], the use case of this thesis. More broadly, two fundamental limitations of radiotherapy are unlikely to be overcome: (i) Since X-rays can be directed locally only, radiation is of limited use when cancer has metastasized, and (ii) radiation can induce secondary cancer.

Main findings of this work

Convolutional neural networks (CNNs) are faster than registration with morphons. So even if they do not always yield better results, they are useful.

Even with a very small dataset size (~ 100), reasonable results can still be obtained. This is an unexpected finding in light of the numbers (1 million or 100,000) quoted in other applications.

Knowledge transfer from CT to CBCT works quite well and improves the performance of CNNs without requiring more labeled CBCTs.

With a small dataset, it is better to work at the image level (i.e., apply simple transformations to make training CTs look more like CBCTs) than at the feature level. It is, however, still unclear whether this conclusion would change if more data were available.

A software framework is proposed to help universities create a consortium with light paperwork to train a model together.

Perspectives

In this thesis, we chose prostate cancer as a use case since large inter-fractional variation in the pelvic area makes it a good candidate for adaptive radiotherapy. It would be interesting to apply the proposed methods to other clinical sites that also rely on CBCT for daily imaging (e.g., head and neck cancers).

Interestingly, the quality gap between this modality and CT has been shrinking over the past few years thanks to technological improvement in image acquisition and reconstruction. The methods for CT to CBCT generalization might therefore have less value in the future. At the same time, research has been progressing towards MRI-based radiotherapy [6]. This might raise interest in knowledge sharing between CT and MRI.

A toddler needs only a few examples of an object to recognize other instances that come in different shapes and colors. In contrast, deep neural networks need thousands of samples. This can be attributed to the fact that AI models lack common sense (i.e., a general model of the world). The way that babies develop a model of the world is by constantly observing their surroundings and predicting what is going to come next. When something unusual happens, you can read the

surprise in their eyes: In such instances, the child has updated its general worldview. Copying nature is not always the best strategy to build the best technology. For example, planes have proven to be a more practical approach than other bird-inspired flying machines. However, nature can point to clues. A possible way to develop a common sense for a deep neural network is by designing surrogate tasks. Self-supervised learning, as it is called, is a promising area of research. Similarly, even though the way humans learn remains a mystery, the brain is unlikely to implement anything like backpropagation [12]. Deep learning has already benefited from neuroscience findings in the past and is likely to continue doing so in the future.

The accuracy of the automated contour is not a goal in itself. What is missing in our project is an analysis of the impact that automated contours have on dose. Even that would not demonstrate the interest of daily image contouring; only a randomized trial between control and adaptive patients would.

More generally, future developments will transform the role of radiotherapy. On the one hand, greater prevention efforts, as well as new cancer treatments, might reduce its role. Current active areas of research include the role of the immune system, cancer metabolism, the role of gene regulation in cancer cells, and the role of the microenvironment of cancer cells [124]. On the other hand, technology-driven improvements in treatment conformity (including particle therapy) and novel biological concepts for personalized treatment [11] will reinforce radiotherapy's position as a major tool in the arsenal of weapons against cancer.

Bibliography

- [1] De la nécessité de la science ouverte, en temps de pandémie et bien au-delà. <https://plus.lesoir.be/334411/article/2020-10-31/la-chronique-de-carta-academica-de-la-necessite-de-la-science-ouverte-en-temps?referer=%2Farchives%2Frecherche%3Fdatefilter%3Dlastyear%26sort%3Ddate%2520desc%26word%3Dopen%2520science>. Accessed: 2020-11-9.
- [2] Open access data management. <https://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/open-access-dissemination.en.htm>. Accessed: 2020-11-9.
- [3] Open science principles. <https://www.mcgill.ca/neuro/open-science/open-science-principles>. Accessed: 2020-11-12.
- [4] World health organization - cancer. <https://www.who.int/en/news-room/fact-sheets/detail/cancer>. Accessed: 2020-06-17.
- [5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.
- [6] Sahaja Acharya, Benjamin W Fischer-Valuck, Rojano Kashani, Parag Parikh, Deshan Yang, Tianyu Zhao, Olga Green, Omar

- Wooten, H Harold Li, Yanle Hu, et al. Online magnetic resonance image guided adaptive radiation therapy: first clinical applications. *International Journal of Radiation Oncology* Biology* Physics*, 94(2):394–403, 2016.
- [7] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks, 2014.
- [8] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2017.
- [9] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Tobias Hepp, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. Medgan: Medical image translation using gans. *Computerized Medical Imaging and Graphics*, 79:101684, 2020.
- [10] Ana María Barragán-Montero, Dan Nguyen, Weiguo Lu, Mu-Han Lin, Roya Norouzi-Kandalan, Xavier Geets, Edmond Sterpin, and Steve Jiang. Three-dimensional dose prediction for lung imrt patients with deep neural networks: robust learning from heterogeneous beam configurations. *Medical physics*, 46(8):3679–3691, 2019.
- [11] Michael Baumann, Mechthild Krause, Jens Overgaard, Jürgen Debus, Søren M Bentzen, Juliane Daartz, Christian Richter, Daniel Zips, and Thomas Bortfeld. Radiation oncology in the era of precision medicine. *Nature Reviews Cancer*, 16(4):234, 2016.
- [12] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*, 2015.
- [13] Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. Fast homomorphic evaluation of deep discretized neural networks. In *Annual International Cryptology Conference*, pages 483–512. Springer, 2018.

- [14] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.
- [15] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [16] Christine Boydev, David Pasquier, Foued Derraz, Laurent Peyrodie, Abdelmalik Taleb-Ahmed, and Jean-Philippe Thiran. Automatic prostate segmentation in cone-beam computed tomography images using rigid registration. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3993–3997. IEEE, 2013.
- [17] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.
- [18] Elliott Brion, Jean Léger, Umair Javaid, John Lee, Christophe De Vleeschouwer, and Benoît Macq. Using planning cts to enhance cnn-based bladder segmentation on cone beam ct. In *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10951, page 109511M. International Society for Optics and Photonics, 2019.
- [19] Elliott Brion, Christian Richter, Benoît Macq, Kristin Stützer, Florian Exner, Esther Troost, Tobias Hölscher, and Luiza Bondar. Modeling patterns of anatomical deformations in prostate patients undergoing radiation therapy with an endorectal balloon. In *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10135, page 1013506. International Society for Optics and Photonics, 2017.
- [20] Sébastien Brousmiche, Jonathan Orban de Xivry, Benoît Macq, and Joao Seco. Su-e-j-125: Classification of cbct noises in terms of their contribution to proton range uncertainty. *Medical Physics*, 41(6Part8):184–184, 2014.

- [21] Jinzheng Cai, Zizhao Zhang, Lei Cui, Yefeng Zheng, and Lin Yang. Towards cross-modal organ translation and segmentation: A cycle- and shape-consistent generative adversarial network. *Medical image analysis*, 52:174–184, 2019.
- [22] Kenny H Cha, Lubomir Hadjiiski, Heang-Ping Chan, Alon Z Weizer, Ajjai Alva, Richard H Cohan, Elaine M Caoili, Chintana Paramagul, and Ravi K Samala. Bladder cancer treatment response assessment in CT using radiomics with deep-learning. *Scientific reports*, 7(1):8738, 2017.
- [23] Kenny H Cha, Lubomir Hadjiiski, Ravi K Samala, Heang-Ping Chan, Elaine M Caoili, and Richard H Cohan. Urinary bladder segmentation in ct urography using deep-learning convolutional neural network and level sets. *Medical physics*, 43(4):1882–1896, 2016.
- [24] Xiangfei Chai, Marcel van Herk, Anja Betgen, Maarten Hulshof, and Arjan Bel. Automatic bladder segmentation on cbct for multiple plan art of bladder cancer using a patient-specific bladder model. *Physics in Medicine & Biology*, 57(12):3945, 2012.
- [25] Cheng Chen, Qi Dou, Hao Chen, and Pheng-Ann Heng. Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest x-ray segmentation. In *International workshop on machine learning in medical imaging*, pages 143–151. Springer, 2018.
- [26] Cheng Chen, Qi Dou, Hao Chen, Jing Qin, and Pheng-Ann Heng. Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 865–872, 2019.
- [27] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54:280–296, 2019.
- [28] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in se-

- mantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 6830–6840, 2019.
- [29] François Chollet et al. Keras. <https://github.com/fchollet/keras>, 2015.
- [30] Patrick Ferdinand Christ, Florian Ettlinger, Felix Grün, Mohamed Ezzeldin A Elshaera, Jana Lipkova, Sebastian Schlecht, Freba Ahmaddy, Sunil Tatavarty, Marc Bickel, Patrick Bilic, et al. Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks. *arXiv preprint arXiv:1702.05970*, 2017.
- [31] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.
- [32] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.
- [33] Joseph Paul Cohen, Margaux Luck, and Sina Honari. Distribution matching losses can hallucinate features in medical image translation. In *International conference on medical image computing and computer-assisted intervention*, pages 529–536. Springer, 2018.
- [34] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International journal of computer vision*, 72(2):195–215, 2007.
- [35] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [36] Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui, Arsany Hany Abdelmessih Guirguis, and Sébastien Louis Alexandre Rouault. Aggregathor: Byzantine machine learning via robust

- gradient aggregation. In *The Conference on Systems and Machine Learning (SysML), 2019*, number CONF, 2019.
- [37] Timo M Deist, Arthur Jochems, Johan van Soest, Georgi Nalbantov, Cary Oberije, Seán Walsh, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, et al. Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: eurocat. *Clinical and translational radiation oncology*, 4:24–31, 2017.
- [38] Geoff Delaney, Susannah Jacob, Carolyn Featherstone, and Michael Barton. The role of radiotherapy in cancer treatment: estimating optimal utilization from a review of evidence-based clinical guidelines. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 104(6):1129–1137, 2005.
- [39] Dario Di Perri and Xavier Geets. Backpropagation applied to handwritten zip code recognition. *UCL-IBA UMRI seminar (unpublished)*, 2015.
- [40] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [41] Peter Doshi, Tom Jefferson, and Chris Del Mar. The imperative to share clinical study reports: recommendations from the tamiflu experience. *PLoS Med*, 9(4):e1001201, 2012.
- [42] Qi Dou, Cheng Ouyang, Cheng Chen, Hao Chen, and Pheng-Ann Heng. Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. *arXiv preprint arXiv:1804.10916*, 2018.
- [43] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [44] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

- [45] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*, pages 139–147. Springer, 1992.
- [46] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- [47] Golnaz Elahi and Eric Yu. Modeling and analysis of security trade-offs—a goal oriented approach. *Data & Knowledge Engineering*, 68(7):579–598, 2009.
- [48] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.
- [49] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.
- [50] Claudio Fiorino, Michele Reni, Angelo Bolognesi, Giovanni Mauro Cattaneo, and Riccardo Calandrino. Intra-and inter-observer variability in contouring prostate and seminal vesicles: implications for conformal treatment planning. *Radiotherapy and oncology*, 47(3):285–292, 1998.
- [51] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.
- [52] Yabo Fu, Yang Lei, Tonghe Wang, Sibao Tian, Pretesh Patel, Ashesh B Jani, Walter J Curran, Tian Liu, and Xiaofeng Yang. Pelvic multi-organ segmentation on cone-beam ct for prostate adaptive radiotherapy. *Medical Physics*, 2020.
- [53] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

- [54] Yaozong Gao, Yeqin Shao, Jun Lian, Andrew Z Wang, Ronald C Chen, and Dinggang Shen. Accurate segmentation of CT male pelvic organs via regression-based deformable models and multi-task random forests. *IEEE transactions on medical imaging*, 35(6):1532–1543, 2016.
- [55] Mohsen Ghafoorian, Alireza Mehrtaash, Tina Kapur, Nico Karssemeijer, Elena Marchiori, Mehran Pesteie, Charles RG Guttman, Frank-Erik de Leeuw, Clare M Tempny, Bram van Ginneken, et al. Transfer learning for domain adaptation in mri: Application in brain lesion segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 516–524. Springer, 2017.
- [56] Michel Ghilezan, Di Yan, and Alvaro Martinez. Adaptive Radiation Therapy for Prostate Cancer. *Seminars in Radiation Oncology*, 20(2):130–137, 2010.
- [57] Davide Giavarina. Understanding bland altman analysis. *Biochemia medica: Biochemia medica*, 25(2):141–151, 2015.
- [58] Daniel R Gomez and Joe Y Chang. Adaptive radiation for lung cancer. *Journal of oncology*, 2011, 2011.
- [59] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [60] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [61] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [62] A Haensch, V Dicken, T Gass, T Morgas, J Klein, H Meine, and HK Hahn. Deep learning based segmentation of organs of the female pelvis in cbct scans for adaptive radiotherapy using ct and cbct data. *Int J Comput Assist Radiol Surg*, 13:179–180, 2018.

- [63] Freddie C Hamdy, Jenny L Donovan, J Lane, Malcolm Mason, Chris Metcalfe, Peter Holding, Michael Davis, Tim J Peters, Emma L Turner, Richard M Martin, et al. 10-year outcomes after monitoring, surgery, or radiotherapy for localized prostate cancer. *N Engl J Med*, 375:1415–1424, 2016.
- [64] Annika Hänsch, Volker Dicken, Jan Klein, Tomasz Morgas, Benjamin Haas, and Horst K Hahn. Artifact-driven sampling schemes for robust female pelvis cbct segmentation using deep learning. In *Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, page 109500T. International Society for Optics and Photonics, 2019.
- [65] Joan A Hatton, Peter B Greer, Colin Tang, Philip Wright, Anne Capp, Sanjiv Gupta, Joel Parker, Chris Wratten, and James W Denham. Does the planning dose–volume histogram represent treatment doses in image-guided prostate radiation therapy? assessment with cone-beam computerised tomography scans. *Radiotherapy and Oncology*, 98(2):162–168, 2011.
- [66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [67] Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3D medical image segmentation: a review. *Medical image analysis*, 13(4):543–563, 2009.
- [68] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998, 2018.
- [69] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [70] Yu-Chi J Hu, Michael D Grossberg, and Gikas S Mageras. Semi-automatic medical image segmentation with adaptive local statistics in conditional random fields framework. In *Engineering in*

- Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 3099–3102. IEEE, 2008.
- [71] Yuankai Huo, Zhoubing Xu, Hyeonsoo Moon, Shunxing Bao, Albert Assad, Tamara K Moyo, Michael R Savona, Richard G Abramson, and Bennett A Landman. Synseg-net: Synthetic segmentation without target modality ground truth. *IEEE transactions on medical imaging*, 38(4):1016–1025, 2018.
- [72] Bulat Ibragimov and Lei Xing. Segmentation of organs-at-risks in head and neck CT images using convolutional neural networks. *Medical physics*, 44(2):547–557, 2017.
- [73] Juan Eugenio Iglesias and Mert R Sabuncu. Multi-atlas segmentation of biomedical images: a survey. *Medical image analysis*, 24(1):205–219, 2015.
- [74] Guillaume Janssens, Laurent Jacques, Jonathan Orban de Xivry, Xavier Geets, and Benoit Macq. Diffeomorphic registration of images with variable contrast enhancement. *International journal of biomedical imaging*, 2011, 2011.
- [75] Umair Javaid, Kevin Souris, Damien Dasnoy, Sheng Huang, and John A Lee. Mitigating inherent noise in monte carlo dose distributions using dilated u-net. *Medical Physics*, 46(12):5790–5798, 2019.
- [76] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. URL: <http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey>, 3:1–12, 2008.
- [77] Jue Jiang, Yu Chi Hu, Neelam Tyagi, Andreas Rimner, Nancy Lee, Joseph O Deasy, Sean Berry, and Harini Veeraraghavan. Psigan: Joint probabilistic segmentation and image distribution matching for unpaired cross-modality adaptation based mri segmentation. *IEEE Transactions on Medical Imaging*, 2020.
- [78] Jue Jiang, Yu-Chi Hu, Neelam Tyagi, Pengpeng Zhang, Andreas Rimner, Gig S Mageras, Joseph O Deasy, and Harini Veeraraghavan. Tumor-aware, adversarial domain adaptation from ct to mri

- for lung cancer segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 777–785. Springer, 2018.
- [79] Arthur Jochems, Timo M Deist, Johan Van Soest, Michael Eble, Paul Bulens, Philippe Coucke, Wim Dries, Philippe Lambin, and Andre Dekker. Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital—a real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467, 2016.
- [80] Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, et al. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In *International Conference on Information Processing in Medical Imaging*, pages 597–609. Springer, 2017.
- [81] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3D cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.
- [82] Samaneh Kazemifar, Anjali Balagopal, Dan Nguyen, Sarah McGuire, Raquibul Hannan, Steve Jiang, and Amir Owringi. Segmentation of the prostate and organs at risk in male pelvic CT images using deep learning. *arXiv preprint arXiv:1802.09587*, 2018.
- [83] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020.
- [84] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer cryptocurrency with proof-of-stake. *self-published paper*, August, 19:1, 2012.
- [85] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

-
- [86] S Klein and M Staring. elastix, the manual, 2018. *Available at <http://elastix.isi.uu.nl/download/elastix-4.9.0-manual.pdf>*.
- [87] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2009.
- [88] Hans Knutsson and Mats Andersson. Morphons: Paint on priors and elastic canvas for segmentation and registration. In *Scandinavian Conference on Image Analysis*, pages 292–301. Springer, 2005.
- [89] Simon Kohl, David Bonekamp, Heinz-Peter Schlemmer, Kaneschka Yaqubi, Markus Hohenfellner, Boris Hadaschik, Jan-Philipp Radtke, and Klaus Maier-Hein. Adversarial networks for the detection of aggressive prostate cancer. *arXiv preprint arXiv:1702.08014*, 2017.
- [90] Lars König, Alexander Derksen, Nils Papenberg, and Benjamin Haas. Deformable image registration for adaptive radiotherapy with guaranteed local rigidity constraints. *Radiation Oncology*, 11(1):122, 2016.
- [91] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *arXiv preprint arXiv:1210.5644*, 2012.
- [92] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [93] Jessica L Krok-Schoen, James L Fisher, Ryan D Baltic, and Electra D Paskett. White-black differences in cancer incidence, stage at diagnosis, and survival among adults aged 85 years and older in the united states. *Cancer Epidemiology and Prevention Biomarkers*, 25(11):1517–1523, 2016.
- [94] Nir Kshetri. Blockchain and electronic healthcare records [cybertrust]. *Computer*, 51(12):59–63, 2018.

-
- [95] Philippe Lambin, Ruud GPM Van Stiphout, Maud HW Star-mans, Emmanuel Rios-Velazquez, Georgi Nalbantov, Hugo JWL Aerts, Erik Roelofs, Wouter Van Elmpt, Paul C Boutros, Pier-luigi Granone, et al. Predicting outcomes in radiation oncology—multifactorial decision support systems. *Nature reviews Clinical oncology*, 10(1):27–40, 2013.
- [96] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.
- [97] Måns Larsson, Yuhang Zhang, and Fredrik Kahl. Robust abdominal organ segmentation using regional convolutional neural networks. In *Scandinavian Conference on Image Analysis*, pages 41–52. Springer, 2017.
- [98] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [99] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [100] Kai-Fu Lee. *AI superpowers: China, Silicon Valley, and the new world order*. Houghton Mifflin Harcourt, 2018.
- [101] Jean Léger, Elliott 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, 2020.
- [102] Jean Léger, Elliott Brion, Umair Javaid, John Lee, Christophe De Vleeschouwer, and Benoit Macq. Contour propagation in ct scans with convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 380–391. Springer, 2018.

- [103] Yang Lei, Tonghe Wang, Sibor Tian, Xue Dong, Ashesh B Jani, David Schuster, Walter J Curran, Pretesh Patel, Tian Liu, and Xiaofeng Yang. Male pelvic multi-organ segmentation aided by cbct-based synthetic mri. *Physics in Medicine & Biology*, 65(3):035013, 2020.
- [104] Hongwei Li, Timo Loehr, Benedikt Wiestler, Jianguo Zhang, and Bjoern Menze. e-uda: Efficient unsupervised domain adaptation for cross-site medical image segmentation. *arXiv preprint arXiv:2001.09313*, 2020.
- [105] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.
- [106] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.
- [107] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.
- [108] Fang Liu. Susan: segment unannotated image structure using adversarial network. *Magnetic resonance in medicine*, 81(5):3330–3345, 2019.
- [109] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [110] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105, 2015.

- [111] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.
- [112] Sébastien Lugan, Paul Desbordes, Eliott Brion, Luis Xavier Ramos Tormo, Axel Legay, and Benoît Macq. Secure architectures implementing trusted coalitions for blockchained distributed learning (tclearn). *IEEE Access*, 7:181789–181799, 2019.
- [113] Suhuai Luo, Qingmao Hu, Xiangjian He, Jiaming Li, Jesse S Jin, and Mira Park. Automatic liver parenchyma segmentation from abdominal CT images using support vector machines. In *Complex Medical Engineering, 2009. CME. ICME International Conference on*, pages 1–5. IEEE, 2009.
- [114] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6778–6787, 2019.
- [115] Rui Ma, Pin Tao, and Huiyun Tang. Optimizing data augmentation for semantic segmentation on small-scale dataset. In *Proceedings of the 2nd International Conference on Control and Computer Vision*, pages 77–81, 2019.
- [116] David Mattes, David R Haynor, Hubert Vesselle, Thomas K Lewellen, and William Eubank. Pet-ct image registration in the chest using free-form deformations. *IEEE transactions on medical imaging*, 22(1):120–128, 2003.
- [117] Maciej A Mazurowski, Mateusz Buda, Ashirbani Saha, and Mustafa R Bashir. Deep learning in radiology: an overview of the concepts and a survey of the state of the art. *arXiv preprint arXiv:1802.08717*, 2018.
- [118] Angelo Mencarelli, Simon Robert van Kranen, Olga Hamming-Vrieze, Suzanne van Beek, Coenraad Robert Nico Rasch, Marcel van Herk, and Jan-Jakob Sonke. Deformable image registration for adaptive radiation therapy of head and neck cancer: accuracy and

- precision in the presence of tumor changes. *International Journal of Radiation Oncology* Biology* Physics*, 90(3):680–687, 2014.
- [119] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.
- [120] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.
- [121] Fausto Milletari, Alex Rothberg, Jimmy Jia, and Michal Sofka. Integrating statistical prior knowledge into convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 161–168. Springer, 2017.
- [122] M. Moteabbed, A. Trofimov, G. C. Sharp, Y. Wang, A. L. Zietman, J. A. Efstathiou, and H. M. Lu. Proton therapy of prostate cancer by anterior-oblique beams: Implications of setup and anatomy variations. *Physics in Medicine and Biology*, 62(5):1644–1660, 2017.
- [123] Kana Motegi, Hidenobu Tachibana, Atsushi Motegi, Kenji Hotta, Hiromi Baba, and Tetsuo Akimoto. Usefulness of hybrid deformable image registration algorithms in prostate radiation therapy. *Journal of applied clinical medical physics*, 20(1):229–236, 2019.
- [124] Siddhartha Mukherjee. *The emperor of all maladies: a biography of cancer*. Simon and Schuster, 2010.
- [125] Jakub Nalepa, Michał Marcinkiewicz, and Michał Kawulok. Data augmentation for brain-tumor segmentation: A review. *Frontiers in Computational Neuroscience*, 13, 2019.
- [126] Mizuho Nishio, Shunjiro Noguchi, and Koji Fujimoto. Automatic pancreas segmentation using coarse-scaled 2d model of deep learn-

- ing: Usefulness of data augmentation and deep u-net. *Applied Sciences*, 10(10):3360, 2020.
- [127] Philip Novosad, Vladimir Fonov, and D Louis Collins. Unsupervised domain adaptation for the automated segmentation of neuroanatomy in mri: a deep learning approach. *bioRxiv*, page 845537, 2019.
- [128] Masahiro Oda, Natsuki Shimizu, Ken’ichi Karasawa, Yukitaka Nimura, Takayuki Kitasaka, Kazunari Misawa, Michitaka Fujiwara, Daniel Rueckert, and Kensaku Mori. Regression forest-based atlas localization and direction specific atlas generation for pancreas segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 556–563. Springer, 2016.
- [129] Seungjong Oh and Siyong Kim. Deformable image registration in radiation therapy. *Radiation oncology journal*, 35(2):101, 2017.
- [130] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A Cook, Antonio De Marvao, Timothy Dawes, Declan P O’Regan, et al. Anatomically constrained neural networks (acnns): application to cardiac image enhancement and segmentation. *IEEE transactions on medical imaging*, 37(2):384–395, 2017.
- [131] Chiara Paganelli, Marta Peroni, Marco Riboldi, Gregory C Sharp, Delia Ciardo, Daniela Alterio, Roberto Orecchia, and Guido Baroni. Scale invariant feature transform in adaptive radiation therapy: a tool for deformable image registration assessment and re-planning indication. *Physics in Medicine & Biology*, 58(2):287, 2012.
- [132] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.
- [133] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

- [134] Cheng Peng, Ergun Ahunbay, Guangpei Chen, Savannah Anderson, Colleen Lawton, and X Allen Li. Characterizing interfraction variations and their dosimetric effects in prostate cancer radiotherapy. *International Journal of Radiation Oncology* Biology* Physics*, 79(3):909–914, 2011.
- [135] Christian S Perone, Pedro Ballester, Rodrigo C Barros, and Julien Cohen-Adad. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194:1–11, 2019.
- [136] Daniel F Polan, Samuel L Brady, and Robert A Kaufman. Tissue segmentation of computed tomography images using a random forest algorithm: a feasibility study. *Physics in Medicine & Biology*, 61(17):6553, 2016.
- [137] Floris Pos and Peter Remeijer. Adaptive Management of Bladder Cancer Radiotherapy. *Seminars in Radiation Oncology*, 20(2):116–120, 2010.
- [138] Floris Pos and Peter Remeijer. Adaptive management of bladder cancer radiotherapy. In *Seminars in radiation oncology*, volume 20, pages 116–120. Elsevier, 2010.
- [139] Adhish Prasoon, Kersten Petersen, Christian Igel, François Lauze, Erik Dam, and Mads Nielsen. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *International conference on medical image computing and computer-assisted intervention*, pages 246–253. Springer, 2013.
- [140] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.
- [141] Hariharan Ravishankar, Rahul Venkataramani, Sheshadri Thiruvankadam, Prasad Sudhakar, and Vivek Vaidya. Learning and incorporating shape models for semantic segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 203–211. Springer, 2017.

- [142] Bastien Rigaud, Antoine Simon, Joël Castelli, Caroline Lafond, Oscar Acosta, Pascal Haigron, Guillaume Cazoulat, and Renaud de Crevoisier. Deformable image registration for radiation therapy: principle, methods, applications and evaluation. *Acta Oncologica*, pages 1–13, 2019.
- [143] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [144] Holger R Roth, Hirohisa Oda, Yuichiro Hayashi, Masahiro Oda, Natsuki Shimizu, Michitaka Fujiwara, Kazunari Misawa, and Kensaku Mori. Hierarchical 3D fully convolutional networks for multi-organ segmentation. *arXiv preprint arXiv:1704.06382*, 2017.
- [145] Jan Schreier, Angelo Genghi, Hannu Laaksonen, Tomasz Morgas, and Benjamin Haas. Clinical evaluation of a full-image deep segmentation algorithm for the male pelvis on cone-beam ct and ct. *Radiotherapy and Oncology*, 145:1–6, 2020.
- [146] Gregory Sharp, Karl D Fritscher, Vladimir Pekar, Marta Peroni, Nadya Shusharina, Harini Veeraraghavan, and Jinzhong Yang. Vision 20/20: perspectives on automated image segmentation for radiotherapy. *Medical physics*, 41(5), 2014.
- [147] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.
- [148] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
- [149] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [150] Matthias Söhn, Mattias Birkner, Yuwei Chi, Jian Wang, Di Yan, Bernhard Berger, and Markus Alber. Model-independent, multimodality deformable image registration by local matching of

- anatomical features and minimization of elastic energy. *Medical physics*, 35(3):866–878, 2008.
- [151] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601, 2017.
- [152] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer, 2017.
- [153] Yoshiki Takayama, Noriyuki Kadoya, Takaya Yamamoto, Kengo Ito, Mizuki Chiba, Kousei Fujiwara, Yuya Miyasaka, Suguru Dobashi, Kiyokazu Sato, Ken Takeda, et al. Evaluation of the performance of deformable image registration between planning ct and cbct images for the pelvic region: comparison between hybrid and intensity-based dir. *Journal of radiation research*, 58(4):567–571, 2017.
- [154] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.
- [155] Jean-Philippe Thirion. Image matching as a diffusion process: an analogy with maxwell’s demons. 1998.
- [156] Maria Thor, Jørgen BB Petersen, Lise Bentzen, Morten Høyer, and Ludvig Paul Muren. Deformable image registration for contour propagation from ct to cone-beam ct scans in radiotherapy of prostate cancer. *Acta Oncologica*, 50(6):918–925, 2011.
- [157] Tong Tong, Robin Wolz, Zehan Wang, Qinquan Gao, Kazunari Misawa, Michitaka Fujiwara, Kensaku Mori, Joseph V Hajnal, and Daniel Rueckert. Discriminative dictionary learning for abdominal multi-organ segmentation. *Medical image analysis*, 23(1):92–104, 2015.
- [158] Eric Topol. *Deep medicine: how artificial intelligence can make healthcare human again*. Hachette UK, 2019.

- [159] R Trullo, C Petitjean, S Ruan, B Dubray, D Nie, and D Shen. Segmentation of organs at risk in thoracic CT images using a sharp-mask architecture and conditional random fields. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pages 1003–1006. IEEE, 2017.
- [160] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.
- [161] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [162] AJAJ van de Schoot, G Schooneveldt, S Wognum, MS Hoogeman, X Chai, LJA Stalpers, CRN Rasch, and Arjan Bel. Generic method for automatic bladder segmentation on cone beam ct using a patient-specific bladder shape model. *Medical physics*, 41(3), 2014.
- [163] Viktor Varkarakis, Shabab Bazrafkan, and Peter Corcoran. Deep neural network and data augmentation methodology for off-axis iris segmentation in wearable headsets. *Neural Networks*, 121:101–121, 2020.
- [164] Yi Wang, Jason A Efstathiou, Gregory C Sharp, Hsiao-Ming Lu, I Frank Ciernik, and Alexei V Trofimov. Evaluation of the dosimetric impact of interfractional anatomical variations on prostate proton therapy using daily in-room CT images. *Medical physics*, 38(8):4623–33, aug 2011.
- [165] Elisabeth Weiss, Jian Wu, William Sleeman, Joshua Bryant, Priya Mitra, Michael Myers, Tatjana Ivanova, Nitai Mukhopadhyay, Viswanathan Ramakrishnan, Martin Murphy, et al. Clinical evaluation of soft tissue organ boundary visualization on cone-beam computed tomographic imaging. *International Journal of Radiation Oncology* Biology* Physics*, 78(3):929–936, 2010.

- [166] Ola Westrand and Stina Svensson. The anaconda algorithm for deformable image registration in radiotherapy. *Medical physics*, 42(1):40–53, 2015.
- [167] Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing*, 2019.
- [168] Andrew J Woerner, Mehee Choi, Matthew M Harkenrider, John C Roeske, and Murat Surucu. Evaluation of deformable image registration-based contour propagation from planning ct to cone-beam ct. *Technology in cancer research & treatment*, 16(6):801–810, 2017.
- [169] Yingda Xia, Dong Yang, Zhiding Yu, Fengze Liu, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation. *Medical Image Analysis*, page 101766, 2020.
- [170] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.
- [171] Dong Yang, Daguang Xu, S Kevin Zhou, Bogdan Georgescu, Mingqing Chen, Sasa Grbic, Dimitris Metaxas, and Dorin Comaniciu. Automatic liver segmentation using an adversarial image-to-image network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 507–515. Springer, 2017.
- [172] Junlin Yang, Nicha C Dvornek, Fan Zhang, Julius Chapiro, MingDe Lin, and James S Duncan. Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 255–263. Springer, 2019.

- [173] Xiaofeng Yang, Ning Wu, Guanghui Cheng, Zhengyang Zhou, S Yu David, Jonathan J Beitler, Walter J Curran, and Tian Liu. Automated segmentation of the parotid gland based on atlas registration and machine learning: a longitudinal MRI study in head-and-neck radiation therapy. *International Journal of Radiation Oncology Biology Physics*, 90(5):1225–1233, 2014.
- [174] V Zambrano, H Furtado, D Fabri, C Lütgendorf-Caucig, J Góra, M Stock, R Mayer, W Birkfellner, and D Georg. Performance validation of deformable image registration in the pelvic region. *Journal of radiation research*, 54(suppl_1):i120–i128, 2013.
- [175] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *Unpublished Draft. Retrieved*, 19:2019, 2019.
- [176] Qingchen Zhang, Laurence T Yang, and Zhikui Chen. Privacy preserving deep computation model on cloud for big data feature learning. *IEEE Transactions on Computers*, 65(5):1351–1362, 2015.
- [177] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2030, 2017.
- [178] Zizhao Zhang, Lin Yang, and Yefeng Zheng. Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9242–9251, 2018.
- [179] He Zhao, Huiqi Li, Sebastian Maurer-Stroh, Yuhong Guo, Qiuju Deng, and Li Cheng. Supervised segmentation of un-annotated retinal fundus images by synthesis. *IEEE transactions on medical imaging*, 38(1):46–56, 2018.
- [180] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

-
- [181] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
 - [182] Wentao Zhu, Xiang Xiang, Trac D Tran, Gregory D Hager, and Xiaohui Xie. Adversarial deep structured nets for mass segmentation from mammograms. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 847–850. IEEE, 2018.
 - [183] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.